AL/HR-TP-1994-0038

ARMSTRONG LABORATORY

RESEARCH, DEVELOPMENT, TRAINING AND EVALUATION (RDT&E) SUPPORT

William J. Salter
Mark Burstein
David Getty

BBN SYSTEMS AND TECHNOLOGIES
10 Moulton Street
Cambridge, Massachusetts 02138

HUMAN RESOURCES DIRECTORATE
LOGISTICS RESEARCH DIVISION
2698 G Street
Wright-Patterson Air Force Base, Ohio 45433-7604

December 1994

Final Technical Paper for the Period June 1993 to June 1994

AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-7604

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation, or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this paper and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

ROBERT E. SMTIH, Capt, USAF
Contract Monitor

BERTRAM W. CREAM, Chief
Logistics Research Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1994	Final - June 93 to June 94

4. TITLE AND SUBTITLE
Research, Development, Training and Evaluation (RDT&E) Support

5. FUNDING NUMBERS
C - F33615-91-D-0009
PE - 62205F
PR - 1710
TA - 00
WU - 60

6. AUTHOR(S)
William J. Salter
Mark Burnstein
David Getty

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
BBN Systems and Technologies
10 Moulton Street
Cambridge, MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Logistics Research Division
2698 G Street
Wright-Patterson AFB, OH 45433-7604

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AL/HR-TP-1994-0038

11. SUPPLEMENTARY NOTES
Armstrong Laboratory Technical Monitor: Capt Robert Smith, AL/HRGA, DSN 785-7773

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

There is a growing concern with respect to the eroding manufacturing base of the United States. This paper identifies and evaluates the majority of the state-of-the-art simulation and modeling tools. Also, the paper identifies research areas which will address the shortcomings of present simulation and modeling methodologies.

14. SUBJECT TERMS		15. NUMBER OF PAGES
business process re-engineering	modeling	93
flexible	re-engineering	16. PRICE CODE
lean	simulation	
	total quality management	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

Table of Contents

List of Figures

Page

(This page intentionally blank)

1. Introduction

This document is the Final Report for CABES, Delivery Order 7 on the RDT&E Contract from Armstrong Laboratory, Human Resources Directorate, to BBN, Contract Number F33615-91-D-0009. It is CDRL A007. In this introductory section, we briefly lay out the background of the project and the approach taken, and present an overview of the balance of this document.

After the introduction, the report has three primary sections. **Section 2, "Issues in Computer-Aided Business Engineering and Simulation,"** presents and discusses some issues that should be considered in selecting and using software to support modeling organizations or organizational processes. In **Section 3, "Software Packages,"** we discuss over 30 available software packages in varying levels of detail, providing more information on those directly concerned with simulation. We present a typology into which the software has been organized, discuss the branches of that typology, then provide concise individual discussions of each software package. In **Section 4, "Research and Development Areas,"** we present and discuss research and development for CABES, the primary thrust of this report as defined in the original Statement of Work (SOW).

1.1 Background

Over the past several years, concern has grown in business and government circles about the competitiveness of American industry in an increasingly global economy. Recession and severe business problems have forced many American businesses, including such hitherto stalwart blue chips as IBM, General Motors, and Sears Roebuck, to downsize substantially and to reexamine their business foci and organization. Manufacturing jobs are declining in America, being replaced (although not one-for-one) by lower-paying service jobs.

Meanwhile, a new model of business organization is being touted in the business and popular press, characterized by customer focus, flexibility, and speed in responding to market requirements and other forces; close coordination among suppliers and the business; increased use of advanced information systems; and empowerment of employees to control and manage their own work. The details of this model, largely imported from Japan, vary in different presentations of it, but the basic focus is clear: American businesses are too slow and overstaffed, particularly at middle management levels, and have lost sight of their primary business purposes. To survive and prosper, they must be "lean" and "flexible," concentrating on meeting customer needs with high-value-added activities.

Reflecting the prevalence of this perspective, a book called *Reengineering the Corporation* (Hammer & Champy 1993) was on the *New York Times* bestseller list for over a year; it's first sentence makes the point rather strongly: "The central thesis of this book -- that American corporations must undertake nothing less than a radical reinvention of how they do their work -- may strike some as extreme (p. iv)." The authors go on to argue that their recommendations can be followed, and have been by

a number of reinvigorated companies, and that not to do so is to embrace defeat and to endorse inevitable decline in America's standard of living.

The Department of Defense (DoD) has been experiencing similar pressures, largely brought on by the end of the cold war, substantial cuts in the defense budget, and ongoing efforts at redefining the role of the American military in a one-superpower world. The DoD is concerned with developing a customer focus, with increasing efficiency, while fostering dual-use technology development.

These concerns go beyond the DoD, across all of the government. In early September 1993, "the national performance report" was released. Led by Vice President Al Gore, a team developed a report on "reinventing government." Its title proclaims its mission boldly: "From Red Tape to Results: Creating a Government that Works Better and Costs Less." Of the more than 200 specific recommendations made in the report, dozens are for "reinventing," reengineering," "redesigning," or conducting "bottom-up review."

A set of vague but powerful ideas is beginning to pervade both public policy debate and board room discussions. The objectives cannot be in doubt; working better and costing less are both desirable, whatever the activity, and simultaneously achieving both is even better. But the ways to pursue these objectives are less clear, and, despite some of the rhetoric, there are tradeoffs: people will lose jobs, the nature and geographic distribution of work will change, old skills will need to be replaced, and so will some of the workers with those skills.

Stockholders are being told that "business process reengineering" (BPR) will bring profits in the future to replace the losses of the present; employees are being told that their jobs have been eliminated or moved as a result of BPR; and taxpayers are being told that hundreds of billions of dollars can be saved by BPR and that national health care reform can largely be supported by the results of reengineering the relevant processes and activities.

On the one hand, it is possible to view all this excitement as no more than the latest fad in management consulting, perhaps grown beyond the normal boundaries of such trends. On the other hand, some see BPR as providing the last, best chance to preserve America's role as world economic leader into the 21st century and beyond. What is certain is that a great deal of time and attention is being devoted to BPR and similar ideas, that important decisions and actions are being taken in its name, and that it means different things to different people. We believe that it is neither all hype nor the salvation of the American way of life. It contains some important good ideas, many of them codified common sense, that can help organizations; it can also be misunderstood and misapplied, used to justify foolish and destructive actions, as well.

Fortunately, this report, and the study of which it is a deliverable, does not have to sort out the issues raised above. Its focus, narrower but still rather broad, is:

- to assess the state of the art of computer-based tools and simulations to support BPR and related activities;
- to identify gaps and shortfalls in that state of the art; and

- to recommend a prioritized research and development agenda in this area for the Human Resources Directorate of the Air Force's Armstrong Laboratory.

The Human Resources Directorate has supported a number of projects concerned with the simulation and modeling of complex activities involving human and machine collaboration, and this effort can be seen in that context, in two senses. First, much of BPR concerns information systems, the use of which can facilitate the effective reorganization of work but can also disrupt work practices if not suitably designed and introduced. Therefore, BPR must be concerned with complex human-machine systems. Second, a variety of computer-based tools already exist that have some relevance to BPR, and these must be assessed to determine what needs remain and where Armstrong Laboratory's efforts can have the highest leverage.

The most visible and vocal proponent of BPR is Michael Hammer, president of a consulting firm that bears his name, author of an article in the *Harvard Business Review* that first introduced the concept to the business world (1989), and co-author (with James Champy) of the current best seller on the topic. He may be termed a BPR radical; he believes that, "American managers must throw out their old notions about how businesses should be organized and run. They must abandon the organizational and operational principles and procedures they are now using and create entirely new ones" (Hammer & Champy, p. 2). Hammer and Champy define reengineering as, "the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements" (p. 32). We will also call this position the "clean sheet of paper" approach, from the idea that the business should be redesigned starting with a clean sheet of paper so that no old ideas are brought along. It defines one end of the spectrum of what people mean by BPR.

It takes a brave, or desperate, CEO to embark on such a program. Most BPR efforts are substantially narrower in scope, including many of the examples in Hammer and Champy. Activities that have been the foci of BPR efforts include the purchase order process, the handling of insurance claims forms, the handling of incoming orders to a mail-order company, and the way a company gets bids for components from internal and external suppliers. Now, when a department is forced to experience a budget reduction, they may call their planning of how to deal with that cut "business process reengineering," although they may do nothing different than they would have if forced to take such a cut five years ago. Many efforts called BPR are essentially aimed at developing information systems that can make it possible to reduce costs (usually personnel) and speed up a process.

Thus, the **scope** of BPR efforts can vary widely. More to the point, the scope of efforts to change and improve business operations varies widely, whether or not they are called BPR. **We will be concerned with computer-based tools and systems that can support businesses in efforts to modify their operations or activities that vary substantially in scope.** We are not concerned with the terminology: whether or not efforts are "really" BPR is not important and depends on one's definition.

Several aspects of what is often termed BPR are important in understanding the tasks that computer-based tools must support. Comparison of BPR with its predecessor as a management favorite -- total quality management (TQM) -- is useful in this regard. TQM was directly concerned with quality and took a strong customer focus. (Internal "customers" were also identified for support aspects of a business, such as the mailroom or purchasing.) It directed that the processes that led to quality should be identified, that the "critical few" points of maximum leverage on those processes should be targeted, and that quality improvement should be continuous.

BPR and TQM thus share several important attributes:

- **focus on customers**: whatever the nature of the business activity, meeting the needs of its customers is a critical requirement;
- **focus on process**: both TQM and BPR take the view that processes in the business lead to outcomes; and
- **identify the points of leverage**: this is more important in the rhetoric of TQM; although BPR is supposed to look at the entire process, actual BPR activities often also focus on a few key aspects of the process.

Perhaps the most important difference in the way the two are usually described is in terms of **the nature of change**. As discussed above, the "radical" view of BPR is that change should be dramatic and almost total; an incremental approach will not work. TQM, on the other hand, is explicitly directed toward "continuous improvement," an ongoing concern with incremental change, which can lead to substantial improvements in quality over time. The difference thus seems dramatic.

However, we believe that many activities that take place in business under the guise of BPR also take an incremental view of change. For example, SRI has a consulting practice in BPR. Among the "critical success factors" they identify for BPR projects are "limit the scope of change" and "get early successes wherever possible" (SRI 1993). This is far from the radical Hammer and Champy approach and has the same incrementalist flavor as TQM. Thus, we will not take major or radical change to be an essential feature of the approaches to business problems we address.

Two related aspects of BPR follow from its focus on process. First is emphasis on **looking across organizational boundaries**, both horizontal and vertical. Thus, the traditional ways in which tasks are divided among people and departments, and in which monitoring and decision-making responsibilities are divided from the "front-line" work, must not be allowed to constrain the description, analysis, and amelioration of business processes. Second is concern with **multiple aspects of an organization**, including information systems, job descriptions (and the associated issues of skill requirements and training needs), organizational structure, relationships with suppliers and customers, and the like. Thus, BPR projects -- even those on a small scale -- have the potential to affect many aspects of an organization.

We believe that the current concern with BPR, however it be defined, represents a significant opportunity to improve the ways organizations operate. And we believe

that computer-based tools and systems can contribute to the utility of activities to redesign, reorganize, or, simply, improve the ways in which organizations operate.

We do not have a "religious" position on BPR; we will take a pragmatic approach. Many kinds of activities are or soon will be underway to change the ways businesses and other organizations (nonprofit firms, governmental entities, schools and hospitals, etc.) operate. Many types of computer-based tools and systems are available to help those involved in designing and managing such change. We will attempt to find gaps and shortfalls in those tools in general and to identify, in particular, those where targeted research and development resources can have a high payoff.

1.2 Focus of the Report

This project is focused on computer-based tools that support modeling organizations or organizational processes. As discussed in an earlier document, the focus of this work is pragmatic: "We are primarily concerned with simulation and modeling systems and tools that can help managers and others to make decisions about how aspects of their [organizations] should be changed, designed, and organized" (Salter & Burstein 1993, p. 17). There are a great many ways in which software might be useful to managers, including such diverse applications as Management Information Systems (MIS), factory automation, Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM), support for strategic planning, and many others. Naturally, we cannot consider all software that might be used by managers in planning or running organizations, but we have tried to review some examples of software in various categories beyond pure modeling and simulation. Our goal in looking at such software is to provide a sense of the shape of the overall software world in which CABES software exists, but not to characterize such software comprehensively.

We take **model** to mean the **representation of a system in a different form.** In general, the system being represented will or might exist in the world, and the model will be a simplification of that system, attempting to capture certain features of interest. Which features it captures, and how it captures them, will depend on the purposes of the model, the skills of the modeler, and so forth. Based on this definition, a drawing (say, of an organization chart) is a model; it also covers the way the term is used in medicine, as in "the rat provides a good animal model of disease X," which means that rats can be induced to get disease X in a way that has potentially useful parallels to how humans get it.

A **simulation** is a type of **model that allows the computation of certain relationships** in the model; in general, the hope is that some of these computations will be of use in understanding or predicting the behavior of the system being modeled.

Models and simulations are always **simplifications, abstractions** of what is being modeled. Their motivation is to preserve aspects of interest to the modeler and eliminate other, complicating and not very important, aspects, so that the model is more comprehensible and usable. The particular aspects captured in a model can vary widely, even for the same real-world system, depending on the goals of the modeler.

For example, if a factory is the real-world system, the models developed by an architect, a sociologist, and a cost accountant would be very different and might not readily be recognizable as models of the same factory. This is because the goals of those disciplines vary widely, and their theories and methods are very different.

We intentionally use the terms "organization" and "process" broadly. By "organization" we mean essentially a group that works toward a common set of goals, although the particular goals of the group members will vary. An organization has an existence beyond that of its members; thus, an organization is generally extended in time. The term also implies a certain amount of structure. The structure is expressed in roles, which define the responsibilities and functions of specific job categories. Structure is also expressed in vertical and horizontal dimensions, which helps to structure the organization into meaningful subunits. Vertical distinctions -- such as departments, divisions, business units -- are typically along functional and/or topical lines. "Functional lines" are distinctions among functions such as manufacturing, sales, marketing, accounting, maintenance, research and development, engineering, and the like; they can be divided further, such as accounts payable, accounts receivable, purchasing, billing, and so forth. "Topical lines" include things such as medical specialties in a hospital, academic departments in a university, separate divisions devoted to individual marquees in the automobile industry, industry groups in consulting organizations, product lines in the consumer goods industry, distinct technology areas in a research and development organization, and so forth.

Thus, we intend the term "organization" to cover traditional businesses, nonprofits such as schools and some hospitals, and government agencies and departments, including but not limited to military entities. The term is intentionally relevant at multiple levels; thus, for example, a military service, a military base, a maintenance facility on a military base, and a department in that maintenance facility are all organizations.

The idea of an "organizational process" (which we use instead of the more common "business process" since we want to stress that relevant organizations include more than businesses) has been defined by many people. One of the most useful definitions is Thomas Davenport's, from his oft-cited *Process Innovation* (1993): a process is "a structured, measured set of activities designed to produce a specified output for a particular customer or market. [A process orientation] implies a strong emphasis on *how* work is done within an organization, in contrast to a product whose emphasis on the *what*. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action" (p. 5). Michael Hammer and James Champy (1993) have a view that puts more emphasis on the customer and less on the structure: a process is "a collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer" (p. 35).

Action Technologies takes a somewhat different view. They distinguish among three types of processes: "materials processes" are concerned with the manufacturing and movement of products; information processes are concerned with the movement and transformation of information; and "business processes" are focused on the customer

and consist of "human interactions required to meet customer requirements" (Action Technologies 1993, page 20).

The customer is central to the idea of business process in general, we believe, although Action's approach takes it further than anyone else. Customers for business processes may be internal or external to the organization (thus, for example, the "customers" of a travel department are employees of the business itself). The idea is that they are the people for whom work is being done. This raises an important related issue, that of organizational intent or purpose. Organizational purpose captures the idea that things should be done for a reason. In some cases, activities in organizations are done for reasons of ritual or history, and the real reason -- the reason related to the purpose of the enterprise -- has been lost. Such activities should be targeted for elimination through analysis of the organization's current processes and their purposes.

The essential features of a process include **tasks** or **activities** or **steps** -- the things people actually *do* as part of the process; these consume **resources** to process **inputs**. A major resource to be monitored is **time**. Processes are engaged in order to provide **added value** for **customers**. The **quality** of the output of a process is an important component of the value of that output to the customer. Processes can be enabled (or harmed) by **tools** and **technologies**, particularly including **information and communication technology**.

Note that the process orientation toward organizations differs in important respects from more traditional structural or product orientations. The process perspective is strongly focused on what happens in the organization, what and how things get done. Often, in fact, important processes cut across functional and topical lines in an organization. The process view also makes technologies particularly important since the essential effects of technology are to change what gets done and/or how it is accomplished.

(This page intentionally blank)

2. Issues in Computer-Aided Business Engineering and Simulation

There are a range of software tools and systems to support CABES; these implement a variety of approaches. We compare and contrast a number of the approaches and specific systems in later sections of this report. In this section, we take a step back from specific software to look more generally at ways of modeling and thinking about business and organizational processes. We address three major issues which we believe are quite important in thinking about organizational modeling and in using organizational modeling tools. They are:

- static and dynamic modeling approaches,
- strong and weak methodologies, and
- the importance of common sense.

Each of these is the subject of a subsection; this section concludes with a brief summary that draws together the three issues and makes a larger point: no modeling and simulation approach is "best"; **the approach to be used must be driven by consideration of the problem to be addressed.**

2.1 Static and Dynamic Modeling

The distinction between static and dynamic models is central in thinking about modeling in virtually any domain, not just organizational modeling. A **static model** represents information about a system at one point in time; static models tend to represent primarily but not exclusively structural information. This means they are not runnable; they cannot transform inputs into outputs (although they can show the relationships among inputs and outputs, such as in some types of flowcharts). Graphic representations are ways of presenting and representing information; they are an important form of static model. They can be quite powerful; data-flow diagrams, flowcharts, organization charts, and IDEF diagrams are examples of graphic representations. Note that the idea of building a static model of a process is perfectly reasonable. That is, it makes sense to build a static model of a dynamic process; in fact, it can be quite useful to do so in some circumstances. We discuss some of the virtues and liabilities of such an approach below.

A **dynamic model** is a runnable computer-based model. Given some set of assumptions and inputs, it will produce outputs. A common form of dynamic model in physics or economics is a system of equations, which relates changes in some variables to changes in others, generally over time. Such models focus on modeling transformations of inputs into outputs and do not necessarily imply any causal representation of the processes through which those transformations occur. These are continuous models, where outputs change as continuous functions of inputs. Another type of model, more common in the organizational modeling domain, is discrete-event simulation. This kind of simulation models time as passing in discrete steps, during which changes propagate through relations among modeling elements. Such models also transform inputs into outputs over time, but they carry the additional claim of representing the process through which such transformations occur to some

degree of fidelity. They generally have a stronger sense of "things happening" than do continuous models.

Dynamic models in general must include metrics for evaluating their fidelity and for use in predicting behavior as part of "what-if" scenarios. In discrete event simulations, components can have internal structure and, thus, some can support multiple measures of behavior. For example, a drill press might be modeled as taking inputs of unfinished product at some rate, performing some actions, and yielding a product for the next stage at some rate (with some defect rate). Furthermore, it is run by an operator of some skill class with some hourly wage, and costs some amount per hour to operate. As this example suggests, models of organizational processes potentially let the user determine how **long** a simulated process takes, how **many** of certain things are produced, **how much** labor, materials, and other resources are consumed, and the like; they offer the opportunity to provide dynamic and quantitative information about multiple aspects of business processes.

A variety of terms are used for what we have called static models. Management consultants, who often use a particular modeling methodology and graphic language, talk about building a "model" of the relevant parts of the organization or the process, by which they generally mean a static representation. In the world of workflow technology, the common term is "process map."

2.1.1 Benefits of Static Modeling

We believe that, **for some problems, major benefits can be obtained simply by building such a static representation, making the process and procedures explicit.** In many organizations, actual ways of working (often, in contrast to the prescribed methods in handbooks or procedures manuals) have evolved into complex processes, many of which are often only implicit. Examining and documenting these processes can help in a number of respects.

(1) Documenting a process can help people involved in it to develop a **shared understanding and vocabulary**, which can be quite valuable in efforts to enhance the process. The development of such a common understanding is a necessary step in enhancing a process and can go a long way, in and of itself, in rationalizing some processes. Such a common understanding is sometimes lacking, particularly for fragmented processes or processes which exist for historical reasons. In some cases, in fact, participants may have different ideas about who is supposed to do what, which can lead to confusion and inefficiency.

(2) Such an explicit representation can also serve as a common reference frame when work is being done, **facilitating ongoing communication** about who is doing what, about deadlines, about priorities, and the like. This can enhance both performance and ability to monitor the process by making expectations clear and by reducing the opportunities for confusion and "dropping the ball."

(3) Such efforts often begin by **making the goals of the overall process explicit**, which can help participants to become and remain focused on the

organizational importance of what is being done. For example, the basic goal of a purchasing process is to obtain needed resources for personnel in ways that do not violate contractual or legal constraints. The detailed steps in the process -- which, in this example, would involve approvals and handoffs -- presumably exist to support this larger goal. Putting individuals' activities into the context of larger organizational goals helps the individuals to understand their contributions to the overall mission, which can have important motivational effects. This is referred to as **understanding intent**, which has been shown to be important to performance across a wide variety of task settings.

(4) Such explicit consideration of larger goals can help to reveal activities or actions that may exist for historical reasons but that no longer serve organizational purposes. For example, a copy of every original purchase order may be sent to "accounts payable," even though "receiving" may be reconciling invoices and P.O.s and sending an approval copy of the paperwork to "accounts payable." Thus, simply documenting a process explicitly can serve to **identify some steps that may be eliminated**.

(5) Documenting a process can also help to reveal **aspects of its structure that could be improved**, in two respects. First, for each step, the question of why is it being done should be considered, by both the performers of that step and by those who are responsible for understanding the larger process. This will help to reveal ways in which steps relate to larger goals. Second, certain kinds of inefficiencies can be revealed. For example, some steps may be performed sequentially that could be performed in parallel; this is a common finding in process studies. Bottlenecks or people (or job categories) that are overloaded can be discovered. Simply drawing an organizational chart can be valuable because it may show problems with span of control, depth of management chains, complexities of reporting relationships, and the like.

(6) Documenting a process serves as a (partial) **baseline** with which the changed process -- whether changes were small and incremental or complete redesigns -- can be compared.

These benefits of explicit documentation of a process can be achieved without any simulation capabilities. Indeed, they can be, in principle (and have been, in many cases, in practice), achieved without computer support at all, using paper-based formats for obtaining and displaying information or through a series of steps on whiteboards.

Static approaches are more useful for documenting existing processes and for making incremental modifications to them than they are for the extensive redesign of processes, since they do not permit the detailed analyses of the redesigned processes that should be made before they are implemented. PERT and Gantt charts are two well-established static modeling approaches with strongly linked graphic conventions that are quite useful in planning and managing work. In the domain of software development, a number of (contending) static approaches -- Yourdon, DeMarco,

Bachman, for example -- have been developed, most with a choice of computer-based tools to support them.

2.1.2 Benefits of Dynamic Modeling

Dynamic models offer several important advantages over static approaches. In general, they do this by substantially increasing the leverage that can be obtained by using **performance data**, as the specific points below illustrate. This explicit quantitative aspect of dynamic models is important in setting up an initial model of the current process, in analyzing the current process, and in developing and evaluating alternatives for it. Essentially, for modeling and analyzing the current process, one uses actual data that have been collected on the process; for designing and evaluating alternatives, one uses the simulation (tuned to the current process with actual data) to generate hypothetical results.

Performance data -- both actual and generated -- can be at several levels of aggregation. For example, in a manufacturing facility, there might be data on each station in a process (such as throughput rates, defect rates) and for the process as a whole or for major subprocesses (such as those responsible for subassemblies). A similar example could be found in a service environment, such as processing applications for life insurance: data might be available on the rates of credit checkers, underwriters, and so forth, as well as on the average and distribution of cycle times to get a policy out the door after an application has been received.

(1) Performance data about individual steps make it possible to **calibrate** the model of the current process by test runs that produce outputs that can be compared to aggregate performance data for the organization. That is, given a simulation and relevant data about each step or activity, overall performance data can be computed. If the overall performance matches actual aggregate performance data fairly well, that serves to **verify** the model of the current process. Such calibration not only supports the model builder's confidence in what is being done but can also help significantly to explain and justify the model to both participants and senior executives.

In modeling an existing process, the model generally provides one mechanisms for its own validation. That is, measures can be collected for various points or stages within the process (such as processing rates for various stations, arrival distributions of customers at a drive-in window, time distributions to have a series of x-rays, and so forth). These are used to set parameters within the model. Measures across larger components of the model can then be compared to the comparable measures for the actual system (for the examples above, these might be total throughput, number of customers served per teller per hour, or total medical testing time). Substantial differences in these more aggregate measures suggest tuning either the design of the model or the parameters it uses.

This type of analysis is necessary, but not sufficient, for redesigning a business process. It is necessary because developing a reasonable model of an existing process requires a clear understanding of what is going on, without

which change can be very dangerous. It is not sufficient because effective redesign can sometimes involve more than modifying an existing process to reduce costs or time or rework, or to increase quality, throughput, or risk.

(2) Dynamic simulations make it possible to examine the performance of the process under various **operating conditions** (such as changes in loads, workforce, supply delivery times, etc.). To the extent that actual data exist on such variations, they can help to calibrate the model in detail. To the extent that such data must be produced by the model, certain assumptions must often be made. It is important to document such assumptions explicitly. For example, in a bank check-processing operation, a modeler might assume that under conditions of high absenteeism, available workers increase their throughput by a certain percentage, encouraged by performance incentives. The costs of such incentives must be reflected in the model.

Often, several different basic configurations of operating conditions will be used; these are generally termed "scenarios." Typical examples might be increased workload with no increase in resources, decreased resources with no decrease in workload, delays in shipments from suppliers, or performance of subcontractors.

(3) Dynamic simulations make it possible to **compare** the performance of **alternative process designs**. Such comparisons, even with a well-calibrated model, are not necessarily correct, of course, but they do provide a valuable source of information to use in choosing among alternative designs. In general, it is advisable to compare performance of alternative designs across several alternative scenarios, since some designs may be better than others only in certain circumstances. With some simulations, it is also possible to evaluate the **robustness** of a design to variations in assumptions or operating conditions. This can be a critical aspect to evaluate, since a design that works beautifully under ideal conditions may be very brittle, and thus not well-suited to the actual, variable, operating environment. For example, just-in-time manufacturing systems are highly vulnerable to delays in shipments from suppliers, which is a major reason that successful efforts tend to involve a high degree of integration in both information systems and economic costs and benefits between supplier and manufacturer.

(4) Dynamic simulations help to support the development of **monitoring strategies** for the new process being implemented. They can suggest points at which monitoring should be done and ranges of values to be concerned about.

(5) Dynamic simulations make it possible to **identify certain types of problems that static models cannot pinpoint**. Examples include breakdowns under unusual loads and other operating conditions, problems caused by complex feedbacks in the system, and certain kinds of opportunities for parallelism and job restructuring (since many simulations make it possible to show loads for individual paths and steps in a process model).

(6) Since many dynamic models (particularly discrete-event simulations) support the use of multiple measures, they can help to identify **tradeoffs**, consideration of which is critical in making informed decisions. One example has been suggested above: there may be a tradeoff between performance under ideal conditions and robustness. Alternatively, there may be a nonlinear relationship between costs and desired outcomes. For example, a 20 percent decrease in cycle time might be obtainable with redesign, at no increased costs, while a 40 percent decrease might increase costs by 15 percent. Management must make the decision whether the projected additional benefit is worth the cost; the simulation makes the tradeoff explicit. Or there can be tradeoffs between desired outcomes; thus, a substantial decrease in cycle time might be obtainable at the cost of a small increase in error rates. Again, the simulation cannot make the decision, but should be able to help frame it for management.

The primary focus of this document is on simulations, although we also consider static modeling methods.

2.2 Strong and Weak Methodologies

A **methodology** is a prescribed way of doing things. Methodologies can differ substantially in the amount of constraint they apply. Some simulations and representations are closely tied to certain methodologies while others are not; some methodologies can be implemented with paper and pencil or on a whiteboard, while others require a particular representation or simulation. A methodology is often associated with a particular way of viewing the type of problem being addressed, of what matters and how leverage can be obtained to address the problem. Many methodologies bring with them implicit ontologies, by which we mean categories of entities that matter and that can be modeled in the methodology.

A variety of methodologies have been developed to document processes, most of which are now supported by computer-based tools. In some cases, the tools help to automate methodologies that existed previously. For example, the **Answer:Architect** and **Envision** systems both provide graphical tools that facilitate modeling in a variety of established CASE (Computer-Aided Software Engineering or Computer-Aided Systems Engineering) methodologies. Others make an approach possible. For example, the iThink software from High Performance Systems makes it possible to apply the system dynamics approach developed by Jay Forrester to organizational modeling.

Of course, applying a methodology does not ensure that the information put into it will be correct. If people cannot or do not accurately characterize the processes they are engaged in, for whatever reasons, simply using a well-established methodology will not address that problem. Methodologies vary considerably in how they deal with inconsistency and vagueness; in some, no provision is made for people characterizing the same thing in different ways (such as a purchase order approval process, as viewed by different participants in that process); in others, such differences

are themselves seen as important data; in yet others, methods are suggested to resolve inconsistencies and to determine the "true" characterization.

In Section 3, we consider various methodologies in our discussions of the software packages that implement them. The focus here is on the distinctions between and relative advantages and disadvantages of **strong and weak methodologies**. A **strong methodology** prescribes a particular way of looking at an organization or organizational process and specifies the sorts of things that can be modeled, typically in terms of both the entities and the types of relationships among entities. Often, software that implements a strong methodology presents itself as more than a set of tools; documentation can sometimes have a quasi-evangelical tone, claiming that finally the phenomenon has been understood, and the results of that understanding have been made available. In fact, the term sometimes used for such an approach is "religious," to convey the strength of such beliefs, and the idea that divergence from them may be viewed as akin to blasphemy.

For example, the Action Workflow System is based on the "conversational model" developed by Terry Winograd and Fernando Flores (1986). This model views business processes as fundamentally based on "commitments" between pairs of people where, via a process of "negotiation," one member of the pair (the "performer") agrees to perform some activities to meet certain "conditions of satisfaction" for the other member, the "customer." This approach stresses the interpersonal nature of modern work, an aspect that has perhaps not gotten sufficient attention. The Action approach takes a strong position that **interpersonal factors are important**, which we discuss in more detail when we consider the specific Action software.

A second strong methodology is advanced by High Performance Systems. They offer two products based on the systems dynamics approach developed by Jay Forrester. iThink is intended for business processes. This approach views phenomena in terms of networks of linear equations. The central insight is that many real-world systems contain feedback loops; such feedback loops are particularly difficult for people to understand, and system dynamics provides a powerful way to document and master such feedback and, thus, to substantially increase understanding of the phenomena being modeled. The iThink approach argues that this is the best and most useful way to look at events and phenomena in the world around us: financial, social, scientific, and other types of events. Forrester himself has been an evangelist for this approach -- quite innovative and powerful when it was introduced -- for over 30 years.

Strong methodologies offer two basic advantages over weak methodologies. First, such a methodology generally provides an **explicit sequence of steps or activities that can be followed**, which helps to ensure that models do not make significant omissions from the perspective of the methodology. Second, a strong methodology generally represents a considerable amount of work and experience, and thus may well offer a very **useful structure for understanding** organizational problems.

The risks posed by applying a strong methodology flow from the fact that it represents a particular point of view of what matters in understanding organizational structures and processes, a point of view that has the virtue of being made explicit.

The critical risk is that it may not apply well to the problem at hand, or that it may fail to capture certain important aspects of it. These dangers are compounded by the fact that using a computer-based system, especially one that is well worked out, tends to enhance the credibility of the results, sometimes to the extent of swamping experience and common sense. The very explicitness of the steps in a methodology may make it easy to miss aspects of the problem that the methodology does not address. While a strong methodology facilitates certain sorts of analyses, it makes others more difficult. Sometimes elaborate steps must be taken to model something that has no natural expression in the methodology. (This can happen in weak methodologies as well.)

For example, in the Action Workflow modeling approach, issues such as what information is involved in a step, what the goals of a step are, and how the step is accomplished must all be relegated to textual notes attached to an "Action workflow process map." Similarly, to use iThink, all relationships must be stated in terms of linear mathematical relationships, with no way to express other aspects of processes.

Most of the systems we examined have weak methodological foundations, in the sense that the methodology is based on a more ad-hoc language for developing directly simulatable models. Systems that use object-oriented approaches, for example, tend not to have strong built-in biases about the way different objects are used to model information vs. materials flows, even though in practice there may be some important differences. On the other hand, it may be that this flexibility is a benefit in many cases, since the modeler has more freedom to put in detail as appropriate for a particular problem. What it does not do is provide a framework in which the modeler is encouraged or forced to provide additional detail in particular areas when, in fact, that detail is important to understanding the processes better.

We believe that the issues and problems to be addressed in modeling organizations and organizational problems are numerous and varied; therefore, there can be no single "correct" methodological approach. Any approach must be evaluated in terms of how useful it can be to pursuing the goals of the modeling activity. This relativistic approach is at odds with the very definition of a strong methodology. However, we believe that strong methodologies can serve as quite useful tools in appropriate circumstances as long as they are used carefully.

For example, we have had extensive discussions with an organization which is applying the Action Workflow approach to the processes through which certain corporate documents (10Q, 10K, annual report) are prepared. These processes are characterized by many reviewers of the documents, some with multiple reviews, and complex handoffs from one reviewer to the next. The Action Workflow Analyst, with its emphasis on person-to-person interactions and conditions of satisfaction, is an excellent way to capture the sources of the essential difficulties in these processes: people do not always know when to expect drafts or when they must finish their reviews (both aspects of the "conditions of satisfaction"), nor to whom they should return their comments (captured in the diagram of the process flow).

In the same organization, another group is interested in analyzing and re-engineering the processes in the accounts payable department. Although those processes also

involve people, the critical aspects of them concern the information flows within, into, and out of accounts payable, and the uses to which that information must be put. Thus, the Action Workflow Analyst will not naturally capture essential aspects of the processes; therefore, a different modeling tool will be used.

2.3 The Importance of Common Sense

Often, once a solution has been designed and implemented, perhaps after considerable analysis and effort, critics (and, occasionally, supporters) describe it as "nothing but common sense." Such common sense is generally easier to apply in retrospect than in prospect; that is, it is much easier to characterize an approach as just common sense once it has been designed than it is to develop that approach by the application of common sense in the first place. In fact, one of the major virtues of specific methodologies and computer-based tools is that they force explicit representations to which common sense may then be applied. We believe that common sense does, indeed, supply important criteria for looking at organizational solutions.

In particular, it should be possible to justify an organizational approach, once it has been developed, by appeal to common sense; people being asked to implement, and to live with the results of, that approach should be able to understand it and its rationale. They might not all agree with it, to be sure, especially if jobs are lost or if time-honored ways of doing business are changed, but they should be able to make sense of it.

Of course, one of the most valuable contributions that a modeling activity can make is to discover and document important phenomena that are not accessible to common sense. Complex systems in general, especially, but not only those with feedback loops, benefit from thorough, systematic analysis. Common sense is still necessary, of course, but it must be empowered by tools and techniques scaled to the problem. Nevertheless, we strongly believe that both organizational problem-solvers and those who will implement and function within the new processes should step back and consider the new approach from the perspective of common sense. Several questions should be asked:

- How will this affect our customers (internal or external)?
- What is most different about this way of doing things? What's changed?
- What makes it better than what it is replacing?
- Why didn't we do this before?
- What does this make harder to do? Easier?
- How will this affect how I do my job?
- How will this affect how we work together?
- How will this affect required employee skill mixes, performance measurement, compensation policies, training?
- How will we know if it is working? If it isn't working?

2.4 Summary and Conclusion

There is a wide variety of approaches that can be used to model organizations and organizational processes. They vary on a number of dimensions, including complexity, level of detail, ease of use, and cost, as well as in more methodological respects. Some approaches are based on strong methodologies which prescribe world views as well as steps to be followed to do the modeling; others take a much more *laissez-faire* approach, providing a range of tools and some guidance on how those tools might be useful. Regardless of the approach and methodology, common sense must be applied to organizational modeling and to using the results of such modeling.

Two linked themes that run through these ideas help provide a context for our approach to this work and, indeed, for using organizational modeling tools in general. First, any business and almost any other organization must take a **pragmatic orientation** toward such modeling. At a minimum, there must be some sense of why it is being done and what problems it might address. At a maximum, there can be a sharply defined organizational problem that must be solved: it takes too long to bring a new product to market, we must reduce costs in accounting, we are losing customers because it takes too long to process orders and there are too many errors, we must shift toward bidding on maintenance jobs away from reliance on a captive market, our operating budget has been cut with no reduction in goals, we must integrate a new acquisition, we must decide whether to build a new plant or expand the existing one or subcontract more, we must decide how to upgrade our computer systems, and so forth.

This pragmatic orientation has two implications. First, it is important to **frame and bound the problem**. The more sharply it can be defined, the more directly it can be approached. However, this definition itself can be an important aspect of problem solving and can sometimes best proceed in an iterative fashion with involvement of a number of individuals. But modeling tools need a fairly well-defined and bounded problem to be useful, since the first question that must be answered is what is going to be modeled. Second, ultimately, some **decisions** must be made, and decisions almost invariably involve **tradeoffs**. Modeling tools can be particularly useful in helping decision-makers assess alternatives in terms of tradeoffs -- between cost and quality, between reliance on one supplier (and consequent risk of disruptions in supply) and complexities of dealing with multiple suppliers, between reduced costs in support functions and the reaction from professionals of pushing some functions onto them, or between the loss of management control from eliminating a layer of management and the reduced costs and possible increased benefits of empowering workers. Modeling or simulation systems cannot make any of the decisions, but they should be used to frame the tradeoffs.

The second major theme is that there is no one best approach; **different modeling methods and tools are suited for different kinds of problems**. It is beyond the scope of this project to develop a detailed analysis of the types of problems, but several dimensions on which they vary can be identified:

- **size**, meaning the number of people, dollar volume, time frame for change;
- **scope of change**, meaning the approximate scale of improvements desired;
- **severity** of the problem;
- **data availability** to model relevant aspects of the problem;
- **nature** of the problem;
- **criticality**, meaning the relationship of the problem being addressed or the process being modeled to the organization's central mission (thus, for example, purchasing would tend not to be highly critical, while customer service or new product development would);
- **organizational scope**, the range of organizational boundaries being crossed;
- **technological content**; and
- **domain**.

For example, for well-understood transaction-based work processes where managing the human interactions is the key problem, the Action Workflow System might be appropriate. For similar processes where the key problems have more to do with managing and coordinating flows of information, Action would not be appropriate, and other workflow products such as ProcessIT or FlowPATH might be strongly preferred. The software packages have different orientations toward what is essential and, thus, have different capabilities in terms of what is easily modeled. Those capabilities must be matched to the aspects of the problem to use the software effectively.

As another example, consider the use of the Metasoft products by Shawmut Bank. The success of that application was due primarily to the fact that very good data were available on the overall process being analyzed and on the relevant components; the Metasoft tools were able to support building simulations using those data, but a number of other packages would also have worked effectively. The essential requirement is that the package be able to model qualitative relationships in a flow, which many can do. Thus, again, the attributes of the problem must be matched to the software, although in this example there are a number of packages that would have fit well with the problem's requirements in terms of the simulations performed.

If the goal is to improve performance by ten or twenty percent, an incremental approach (which can be supported by a variety of tools, depending on the details of the problem) should be taken; if the organization is facing disaster unless it changes radically, a much more dramatic approach must be taken, in which simulation and modeling may have a place, but leadership and vision will have a far bigger role to play.

(This page intentionally blank)

3. Software Packages

In this next section of the report, we discuss approximately 30 software packages. These packages provide a range of tools and functionalities, and vary considerably in their direct applicability to the simulation of organizations and organizational processes. First we present a typology we have developed to organize the discussion of the individual packages. No claim is made that it provides a definitive means of classifying all possible software that can support organizational modeling. Rather, our aim is more modest and pragmatic: the typology is intended to help the reader, and the possible user of the software, to understand the range of tools available for various problems associated with organizational modeling. The typology is *not* driven by a strong theoretical perspective.

The typology is presented in Figure 3-1 below. In the balance of the introduction to this section of the report, we discuss its organization and the meaning of the various branches.

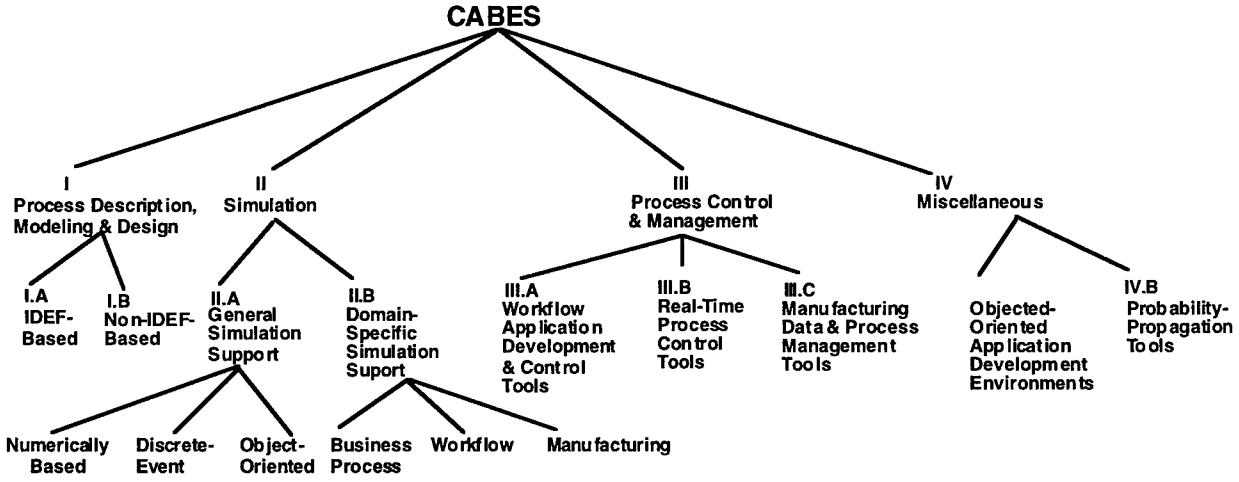

Figure 3-1: Product Taxonomy

The first division of the typology is along the dimension of **stages in the redesign and implementation of new organizational processes.** We divide this, the "organizational engineer's" task, into three major stages, each of which corresponds to a major branch of the typology: (1) description of the existing and proposed new organization's processes, using static modeling techniques; (2) evaluation of these designs using simulation, and (3) ongoing control and management of organizational processes. There is also a fourth branch: miscellaneous software. We briefly discuss each of these major branches of the typology and their subdivisions before considering specific software packages.

(I) **Process Description, Modeling and Design:** This branch of the typology contains tools that support process description and static modeling. Process design is viewed here essentially as a description of a process that does or

does not yet exist. The tools in this category do not have inherent simulation capabilities, although some of these tools or similar ones do exist in product suites with simulators. The processes they describe cannot be animated directly, although the very act of describing processes at the level of detail that these tools promote may still be one of the best means of gaining insight into what is wrong with a process.

The tools described here are typically graphical tools for building labeled graphs of action or work flows, precedence, authority, and data object relationships. They vary in style due primarily to the assumptions about the methodology for which they were developed. We divide them into two types:

(A) IDEF-based tools, which support a form of the IDEF description formalism (usually but not exclusively IDEF0); and

(B) non-IDEF-based tools, which support other description methodologies.

(II) **Simulation:** This branch contains the simulation tools and is the major focus of our analysis. We have subdivided it into two major subbranches, each of which is further divided.

(A) **General simulation support** consists of tools and systems that are intended to support simulations in a variety of domains. These simulations are of three primary types:

(1) **numerically based simulations**, which support the development of equation-based simulations of continuous and discrete processes;

(2) **discrete-event simulation development systems**, primarily based on Petri-net models and the development tools built around them support simulations of material or information flowing through a series of states or stations, over time; and

(3) **object-oriented simulation development environments**, which grew out of but differ in two important respects from the preceding, are based on object-oriented technology, which provides a particularly good match for discrete-event simulation, and are more general simulation-development environments. Thus, they are in no way specialized for simulations of organizations or organizational processes.

(B) **Domain-specific simulation support** consists of tools designed for a specific domain of simulation. We distinguish three primary domains:

(1) **business process simulation**, which contains the tools specifically intended to model business processes, are often adapted or extended versions of tools from manufacturing;

(2) **workflow simulation**, which contains tools that model flows of work as mediated by communications mechanisms (these tools are *not* parts of

suites of workflow application development tools, which are discussed in a later branch of the typology); and

(3) **manufacturing simulation**, which is intended specifically for applications in the manufacturing sector (we have intentionally *not* focused on manufacturing in this project, although we do review briefly a number of projects targeted toward manufacturing).

Manufacturing simulation has two subtypes of tools in this subbranch of the taxonomy:

(i) **general manufacturing simulation environments**, which enable the user to build models in a range of manufacturing domains (some of which also support additional software models that provide an industry-specific layer or library); and

(ii) **industry-specific simulations,** which are narrowly targeted at particular industries.

(III) **Process Control and Management:** A number of software tools support the implementation and ongoing management and control of organizational processes. We have divided them into three types:

(A) **workflow application development and control tools**, which support the development of workflow applications that can run on PCs, enabling computer-based work to be routed and monitored automatically among people;

(B) **real-time process control tools**, which support the monitoring and control of actual processes, typically in a manufacturing setting; and

(C) **manufacturing data and process management tools**, which support the development of MIS and other manufacturing support tools (such as materials and inventory management, customer order management, etc.).

(IV) **Miscellaneous** software has some relevance to organizational modeling but does not fit into the scheme developed above. This consists of two subbranches, the first of which includes the most broadly applicable software we have looked at and the second of which includes some of the most sharply focused.

(A) **Object-oriented application development environments**, derived from rule-based systems tools of the 1980s, provide a general suite of tools to develop applications; these environments are not particularly well-suited to develop simulations.

(B) **Probability-propagation tools**, which run in conjunction with spreadsheets, make it possible to attach uncertainties to particular cell estimates and have those uncertainties propagate to the cells derived from the original cell.

We use the categories outlined above to structure our discussions of the specific software products in the following sections.

3.1 Process Description, Modeling, and Design

Virtually all the tools that do simulations and most of the development tools for execution systems (e.g., workflow, real-time control) provide some form(s) of graphical description tools as part of their packages. However, we have called out some of these design tools separately here, when appropriate, because of their distinguishing features or strong methodological basis, or because they are actually sold separately.

Indeed, most of the so-called CASE tools are of precisely this form: graphical description tools supporting "languages" that are based on a methodology whose strength is derived from making explicit particular aspects of organizational processes. It is generally assumed that by making these properties and relationships explicit, people can discover and repair problems and inefficiencies.

3.1.1 IDEF-Based Modeling Tools

One of the best known and most popular standard graphical languages for process description, IDEF0 provides a simple yet powerful language for decomposing processes into hierarchically organized precedence graphs. Figure 3-2 below shows the basic structure of IDEF diagrams. Each "page" of the model shows a diagram like that shown in Figure 3-2. These diagrams are limited in complexity (usually six or fewer boxes, typically cascaded), and there are only four types of arrows (inputs, controls, outputs, and mechanisms (ICOM)) Each box can itself represent another diagram or "page" of similar complexity, with each of the four basic arrow types from the higher-level diagram feeding into the lower-level one.

This type of model has several advantages:

(1) they are easy for people to read and learn to build;

(2) if used properly, they can capture much of the relevant detail of a process and help modelers to identify problems and inefficiencies;

(3) they lend themselves to activity-based costing analysis; and

(4) they can be used to build simulations, as Meta Software has done.

They also have a number of potential pitfalls:

(1) the basic structure of the diagrams has little inherent semantics, which means that different people may well build very different models of the same process;

(2) the diagrams tend to make representations of feedback and conditional or decision processing somewhat difficult to express;

(3) parallelism is not clearly modeled (although it can be derived from the data and control flows, to some degree); and

(4) producers and customers are not clearly denoted.

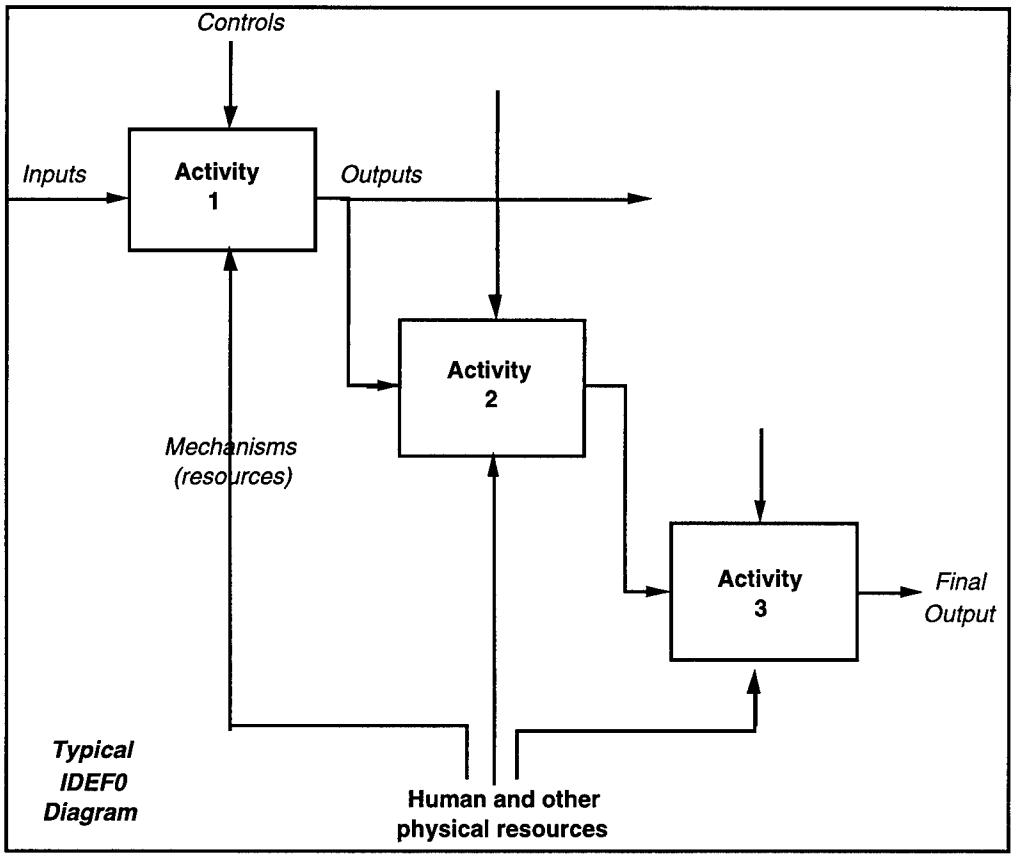

Figure 3-2: IDEF0 Diagram

The limited number of blocks one is supposed to use on a single diagram is both a strength and weakness of these models. As with good programming style, the practice of hierarchically decomposing into small, comprehensible chunks is often a strong plus in terms of maintaining understandability and accuracy. However, it is not always possible to model existing processes that have evolved over time with strict adherence to guidelines about hierarchical decomposition. There is an inherent potential for "over-hierarchicalization," which can bury some important details of a process in ways that make it hard to see redundancies, or can make it difficult if not impossible to show auxiliary information pathways that may, in fact, be important to how things really work.

BPwin (Logic Works Inc.)

Release date: 1993 Approximate cost: $1995 Platforms supported: PC/Windows

BPwin is a graphical modeling tool, based on IDEF0 methodology, for breaking down a business process into its subprocesses and activities. From a Model Definition Editor, the user first defines the process to be analyzed, then creates hierarchical decomposition diagrams using simple point-and-click methods to define component activities of the process and their relationships, using IDEF0 elements. The user interface is intuitive and fairly easy to use. The supplied manual consists largely of informative tutorials. The software provides automated or manual arrow routing and activity numbering. It also provides referential integrity, so that changes made at one level of the model description are automatically cascaded to both parent and child diagrams. This software also supports Activity-Based Costing, in which activities can have costs associated with them and various types of cost summaries and cost breakdowns by subprocess or activity can be reported. In addition, the user can select and customize a number of types of other reports, including a Model Report, Activity, Activity Breakdown Report, Model Consistency Report, and an Arrow Report. The program supports DDE (Dynamic Data Exchange) exporting of reports to other applications, but not OLE (Object Linking and Embedding).

ERwin (Logic Works Inc.)

Release date: 1993 Approximate cost: $1995 Platforms supported: PC/Windows, Macintosh, UNIX workstations

ERwin is very similar in look and feel to BPwin but is designed to model data structures and relational databases rather than business processes. The product allows a user to design database applications with advanced client/server features including primary and foreign keys, indexes, referential integrity constraints, triggers, and domain constraints. The graphical user interface allows the user to construct an Entity-Relationship (ER) model of the (business) rules governing the data in the user's application, in which all entities, attributes, relationships, keys, and index indicators are represented in the diagram according to IDEF1X conventions. Various aspects of the model can be edited, viewed, and printed in a variety of ways, from dictionary reports to high-level model descriptions to detailed SQL (Structured Query Language) database design. Once the ER diagram is complete, ERwin will automatically convert the model into SQL statements which will create the relational database corresponding to the model. The program supports the SQL syntax of a number of different server applications. An important additional feature of the product is the ability to reverse-engineer an existing relational database. ERwin allows the user to take an existing database and automatically construct the corresponding logical ER model. This model can then be edited, upgraded to include new or more advanced features, or merged with other models; a new relational database can then be constructed from the revised model.

Design/IDEF (Metasoft Inc.)

Release date: 1993 (version 3.0), originally released 1987 Approximate cost: $3995, 1-5 copies. Number sold: in the thousands Source language: C Platforms supported: Macintosh, PC/Windows, UNIX X-windows platforms

A high-quality IDEF graphical modeling tool, Design/IDEF supports IDEF0 (process flow), IDEF1 and IDEF1X (data flow), and Entity-Relation diagramming. It supports the government's IDEF0 FIPS (Federal Information Processing System) standard for process and data modeling. Supports import/export to Integrated Definition Language (IDL), Activity Modeling Language (AML), Structured Modeling Language (SML), and KnowledgeWare's Applications Development Workbench (ADW).

Design/IDEF is the front end used for modeling business processes that are to be simulated with Metasoft's Design/CPN (Colored Petri-Nets) simulation system. Together, these products are part of Meta Software's Workflow Analyzer product (see the section on Discrete-Event Simulations).

Design/IDEF has also been very successful as a stand-alone product and has been used throughout the U.S. and Europe. It has also been selected as the tool supplier for the business process redesign for the DoD I-CASE contract, awarded November 1993.

CABRE AI0 and ProCap (AT&T ISTEL)

Release date: 1986 Source language: FORTRAN 77, C Platforms supported: PC(OS2), DEC VAX(VMS), UNIX workstations

CABRE (Computer Aided Business Re-Engineering) is a collection of tools for capturing, diagramming, modeling, and simulating business processes. We focus here on two IDEF-based tools: AI0 for functional process modeling and ProCap for process description and flow modeling.

AI0 aids the user in creating a functional analysis of a business process using the IDEF0 methodology. Business functions, such as assembly of a product, are described graphically in a hierarchical model using IDEF0 elements. The product supports activity-based cost analysis.

ProCap is a tool for graphically modeling a business process using IDEF3 methods. The program was designed to collect information about and document processes by describing entities and capturing precedence and causality relations between entities in a form that is natural to domain experts. The user can graphically build a flow diagram of the process that incorporates logical and temporal constraints in the process as well as detailed information about entities and their roles in the process. ProCap uses IDEF0 and IDEF3 for graphical process description.

3.1.2 Non-IDEF-Based Process Modeling Tools

The first two systems here are both based in the CASE world. Each is built on a general purpose graphical modeling tool with a variety of alternative "views" that allow users to build process models using a number of different CASE process description methodologies. The last system here is unique and embodies a very different world view. Action Technologies' workflow modeling tool is based on the work of Winograd and Flores (1986), who seek to describe work almost entirely in terms of interpersonal communications.

ANSWER:Architect (Sterling Software)

Release date:
Approximate cost: $3000 (baseline system)
Number sold:
Source language:
Platforms supported: PC/Windows, OS/2

This product is a general purpose process description tool that consists of two parts: a base-level system and loadable libraries called CASE-ettes. The base system, which is independent of any particular process description methodology, provides a number of general capabilities for graphically creating and modifying objects and process flow diagrams, entity-relationship diagrams, organizational and structure charts, state-transition diagrams, and reports of many types. Objects and relationships are hierarchically defined with a point-and-click editor, with automatic rerouting of connectors as objects are moved, and multilevel zoom within any diagram. The system provides the ability to create reusable libraries at the model or component level, in an integrated, shareable, LAN-based (Local-Area Network-based) repository with SQL access.

CASE-ette modules, built by Sterling or third parties, customize the system to use a particular process description methodology or technique. Each CASE-ette includes established rules, notation, and terminology for a specific methodology or technique. Modules are available for IBM's LOVEM (Line of Visibility Methodology), ANSWER:Method, Information Engineering, Stradis, Structured, Merise, Foundation/1, and others. LOVEM/CABE (Computer-Aided Business Engineering), for example, is a process flow diagramming methodology that emphasizes customer interactions and opportunities for parallelizing sequential flows.

Envision (Future Tech Systems)

Release date: 1990
Approximate cost: $7500
Number sold: 900
Source language: C, C++
Platforms supported: PC/Windows, OS/2

Envision is an object-oriented process modeling tool that provides support for a set of common CASE-based process description methodologies, while allowing the user to enhance or tailor the graphical notations and rules to fit a particular application. Methods supported include Yourdon, De Marco, Gane & Sarson, ER diagrams, State Transition diagrams, and several others.

The graphical editor provides for object definition, where each object may contain data, child diagrams, and attributes. The system supports multiple diagramming techniques, rules definition, spreadsheet-style matrices, zooming of diagram parts and subparts, and association of a formula and/or data attributes with any defined object or set of objects. The system provides dynamic links to other applications and supports a multi-user, object-oriented networked repository which allows a group of users to share models and project information with enforced data integrity. The product includes a reporting system that supports the production of reports including specification sheets, diagrams, and embedded text files from point-and-click dialogues with the user.

Action WorkFlow Modeler (Action Technologies)

Release date: 1993 Approximate cost: $500 (Analyst; diagramming tool) Number sold: Source language: Platforms supported: PC/Windows, OS/2

As mentioned in Section 2, Action Technologies approaches workflow diagramming from a strong methodological point of view. Their method assumes there is always a fundamental dialogue and relationship between a customer and a performer driving any work that gets done. For each cycle (shown as ovals in Figure 3-3), there are four primary parts: a preparation or proposal phase, in which the work is identified and described by either the customer or the performer; a negotiation and agreement phase where a commitment (implicit or explicit) or contract is made to perform some work; a performance phase; and finally an acceptance phase where the customer takes delivery or rejects the work product.

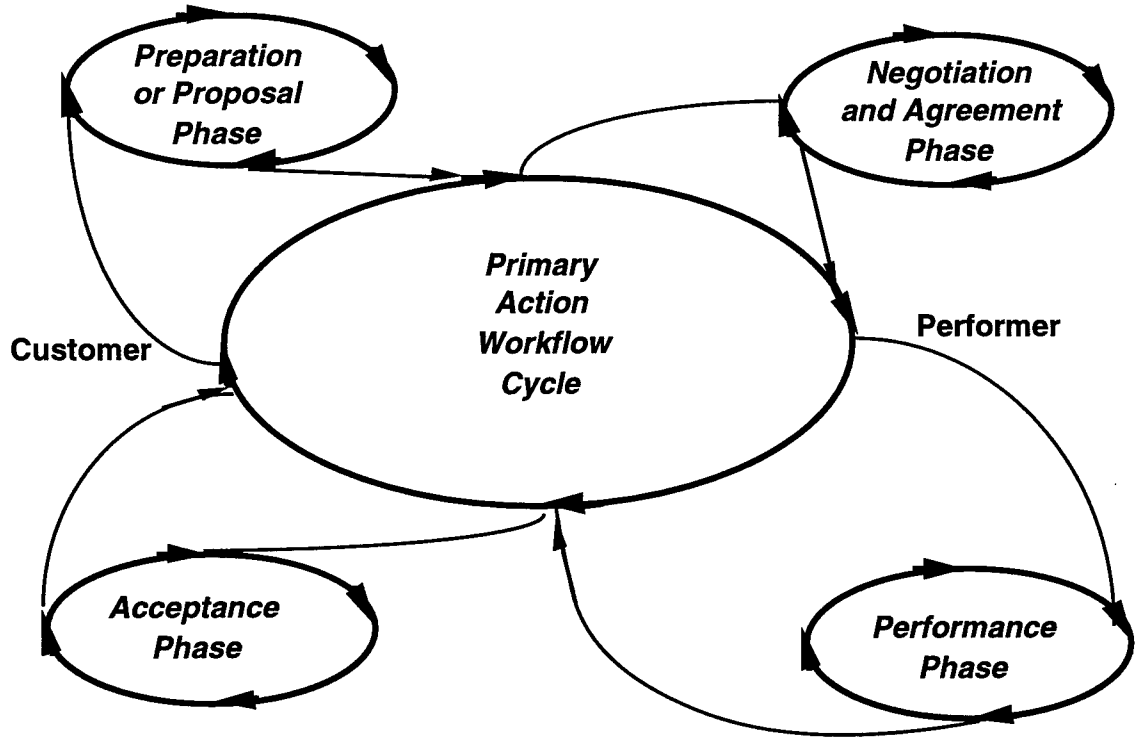

Figure 3-3: Action Workflow Diagram

Secondary flows, which "hang off" the primary workflow, also have a customer and performer. They give more details of the steps done in each phase of the primary workflow. For example, there might be a step in a subordinate workflow for the proposal preparation phase of a contract where the legal department reviews a proposal draft. In that case, the performer would be the legal department, and the

customer the proposal preparer. These subordinate cycles may be chained together or parallel but must ultimately relate back to the primary flow.

Action's workflow diagrams have the benefits and disadvantages of their strong methodological bias. They are useful in that they help to make explicit the critical communication junctures in workflows and who is responsible for performance and acceptance. As such, they can help organizational designers understand the interpersonal processes involved in work.

Conversely, this style of diagramming makes no attempt to support more typical process descriptions, especially when multiple steps are being performed by a single person or group. The whole focus is on flows of communications between customers and performers, rather than the more traditional view where the person(s) doing the work may be incidental to the process or data-flow descriptions.

3.2 Simulation Tools

A primary focus in our work has been on studying the state of the art of simulations that can be used in support of business process re-engineering and redesign. The existing commercial marketplace for this type of software can be divided into those products that are more or less general purpose simulation development tools and those that have been developed specifically for certain types of industrial applications. Further divisions can be made based on the type of the underlying simulation software technology, be it token-passing, object-oriented, or numerically based simulation. Virtually all the products listed here are essentially discrete-event simulations, except those noted as (also) being capable of modeling continuous phenomena.

Many of the general-purpose simulation development environments actually have a suite of "layered" products where the substrate simulation technology is bundled with "objects" or other forms of vocabularies that can be used more directly for building simulations in certain classes of domains. We will describe these layered products separately for the most part.

3.2.1 General Simulation Development Tools

While most simulation tools were originally developed for a particular market, a number have been generalized to a considerable degree so that, in principle, they can be applied to almost any domain for which time is modeled appropriately by the simulator. These tools fall into several categories, based on this factor. Tick-based simulators, such as the numerically based ones, can be used equally well to model discrete event and continuous processes because they do not depend on actions having a discrete beginning and ending time. Discrete event models, in contrast, can be made to run much more quickly, because they will "jump through time" to the next event that needs to be modeled, rather than moving in small, smooth increments through time. On the other hand, if a discrete-event simulator is attached to an animation display system, at least that part of the

system must normally be tick-based in order that the animation does not appear to lurch erratically through time.

3.2.1.1 Numerically Based Simulators (tick-based, discrete, and/or continuous)

This category, as a group, consists of numerical simulations of both discrete and continuous processes. The common focus is on linear and nonlinear difference equations, propagated over unit time intervals, so that the user can observe feedback behavior by viewing X-Y plots of outputs and intermediate products. All of the products support a fairly consistently high level of graphical "wiring" together of the models to be simulated, using a vocabulary of building blocks whose parameters (including the numerical equations relating inputs to outputs over a single time interval) are set via dialogue boxes.

In general, this type of simulation is more appropriate for engineering and scientific simulations than for business processes. However, two of the products (iThink and Extend) make strong claims of supporting business process simulations. They do so in one of two ways. (1) To model individual inputs as being of different types, model builders must encode these variations of inputs and outputs as integers, which are "processed" by conditionals that make reference to that discrete encoding. For example, if orders could be for red or blue cars, the inputs might be 1s or 2s. Building models of this kind is laborious without higher-level supports, and mapping the model to the real situation is not nearly as natural or perspicuous as in a more object-oriented approach, both with respect to developing scenario input data and when viewing outputs. (2) The second style of model (a more typical example in the promotional literature) depicts flows of "paper" or other kinds of requests to perform tasks as numbers that indicate the *size* of an undifferentiated "batch" of objects to process, rather than having an explicit *set* of objects to process. Use of queues in this kind of a model is really a representation of a sequence of batches of various sizes that will be processed at succeeding time intervals. Processing a fraction of those requests differently is modeled by a fraction at the node that distinguishes their breakdown into numbers of outputs (sizes of "batches") to be sent along to different succeeding stations.

In general, this forced choice between two specific modeling styles means that one cannot, at least not nearly as easily as would seem desirable, carry properties of individual objects throughout a process, and it is likely to be unclear to all but experienced modelers what is the correct way to proceed when starting out. In addition, our experiments with iThink indicate that the first style of modeling above, which is much more likely to be the correct choice for models of any complexity, is undersupported with this kind of tool. This style of modeling is *the only choice* in more traditional discrete-event simulation systems and is therefore better supported. On the other hand, if one *can* represent the subject business process as a flow of quantities of uniform kinds of objects and use equations to model both alternative branches in processes and variable processing time, these tools may be sufficient to do some reasonable amount of visualization and process redesign. If this is not clear from the outset, however, a more powerful tool is probably desirable.

iThink (High Performance Systems)

Release date: 1993
Approximate cost: $700
Number sold:
Source language: C
Platforms supported: Macintosh

iThink is one of the more popular tools in this category, from a price and ease-of-use perspective. It is a well-structured package, based on principles of *system dynamics*, that makes the development of models involving small tens of objects relatively easy. Basic building blocks include flows, queues, and numerical models of activity cost in terms of time. Control is modeled by separate nodes and arcs in the graphical model language, unrelated to the flows.

iThink makes strong reference to their rigorous methodology for model construction; however, actually building business process models using their language is quite awkward because much of the control representing decision-making going on within the processes modeled had to be done outside the basic flow model. Reviews indicate that there is a fairly steep learning curve in getting complex models going with iThink. For business process modeling, we believe that much of this arises from the objections cited above for this group of products as a whole.

Extend (Imagine That)

Release date: 1993
Approximate cost: $700 ($1000 for Extend+Manufacturing)
Number sold:
Source language: ModL (a modeling language built on top of C)
Platforms supported: Macintosh

In underlying technology and market orientation, this product is very similar to iThink. However, it takes a somewhat more intuitive approach to model development and provides a greater variety of basic building blocks, making simple business process simulations relatively easy to construct. However, it likely suffers from some of the objections given above for use of this style of product for modeling relatively complex business or manufacturing products.

This product is the "best of breed" of those in its class that we have examined, in terms of its ease of use, understandability of the models that one can construct, and its model construction interface. It makes the construction of hierarchical discrete-event models easier than with iThink, and its object-oriented design makes it more extensible and customizable. Users are invited to build their own model libraries, and there are layered products consisting of libraries of pre-configured building blocks, such as their libraries for discrete-event, continuous, BPR and manufacturing simulations. Although we were not able to examine the manufacturing and BPR libraries, their descriptions suggest that they have attempted to overcome many of the objections that would make iThink inappropriate for BPR modeling. It may be possible for sophisticated users of Extend+BPR to build reasonable small-scale models, similar to those one might build with CACI's SimProcess/SimFactory, for substantially less money. However, there is some reason to believe that one could not build very large models using this tool.

VisSim (Visual Solutions Inc.)

Release date: 1992
Approximate cost: $500-$7000 depending on configuration, add-ons
Number sold: Over 500
Source language: C
Platforms supported: PC/Windows, Unix

VisSim is a visually programmed mathematical modeling and simulation tool designed for engineering applications. It supports numerical methods for solving nonlinear differential equations, described in terms of wiring block diagrams, much like iThink. It supports over 60 linear and nonlinear function blocks for both digital and continuous systems. It is clearly oriented to engineering applications and does not even attempt to work directly in the business process area. In particular, it **does not support discrete-event simulations.** Support for graphical data visualization of simulation products seems particularly strong.

3.2.1.2 Discrete-Event Simulation Development Systems

The underlying simulation technologies represented in this group have been around for a number of years. Discrete-event simulation systems of this kind are essentially all based on some kind of marker or token passing through an event network, where each event along the path is modeled as taking some time that may be determined stochastically. Conditional branching and looping, queuing, and other basic constructs are all supported. They also provide some mechanism for annotating the tokens with attributes that can be changed as the token moves across an event arc[1] or enters a state node.

All these systems are essentially simulators for Petri-nets (Petri, 1966). Petri-nets were originally formulated as condition-action network models where unlabeled tokens were used to mark transitions from one state to another. All these simulation systems (including many of the object-oriented ones in this and the next category) are essentially variants of Colored Petri-Nets (CPN), where the tokens are actually records or objects with properties.[2] Each of these simulation systems shows how information and/or materials traverse a network of states and/or stations. Each has a graphical model development interface that provides for some graphical description of a process as a network and some amount of analysis support environment but is relatively capable of looking at resource-constrained materials flow problems.

One of the advantages of Petri-nets is that there is a precise and explicit semantics to the models. It is not too difficult for a reasonably computer-literate person to understand that semantics, although the step from there to applying it to modeling a real enterprise may still be quite daunting for most. As such, we see products in this group as not really being tools that naive users could use on their own.

---

[1]The nodes and arcs in the graphs traversed in these simulations mean either states and actions, respectively, or actions and states, respectively, depending on the notation of the product developer. We will talk about arcs as being actions here because that is the more common model within this technology group.

[2]In fact, both Workflow Analyzer and ProTEM are explicitly CPN-based, while a recent paper (Taqi et al., 1992) described how the SLAM-II language was equivalent to petri-nets. A review of literature on that system reveals that tokens have and rely on attributes at condition and action points in SLAM-II as well.

SLAMSYSTEM (Pritsker Corp.)

Release date: 1991 (SLAMSYSTEM)
Approximate cost: $9500
Number sold: over 5000
Source languages: FORTRAN (probably C for graphics package)
Platforms supported: IBM PC or compatible with Windows

While perhaps not the progenitor of all of today's simulation tools, the SLAM Simulation language has certainly been in existence the longest of all the systems we reviewed here (Pritsker Corporation was founded in 1973). Written originally in FORTRAN 77, the SLAM II language on which their models are still built provides all the basic constructs for condition-action discrete-event simulation with plenty of hooks so that users can augment the simulation with their own code as needed. SLAMSYSTEM is Pritsker's graphical simulation development environment for Windows. It produces models that can then be run on other platforms without recompiling.

While this tool suite has clearly been used for a variety of real applications of many different sizes, the model development tools are still heavily grounded in the original simulation language. Hence a number of "FORTRANisms," in addition to the idiosynchracies of the SLAM simulation methodology, must be understood to use the tools effectively in a new application, and it is not unusual to have to write additional FORTRAN code to get the behavior that is required. These are significant drawbacks from an ease-of-use perspective.

By virtue of its long history of work in this area, Pritsker has developed a wide variety of clients who have used their simulation tools, often with Pritsker's help, to model a variety of human and machine-centered processes. These include studies of scheduling policies for hospital blocks of operating rooms and associated resources, computer network management problems, fuel storage and distribution plans, physical security staffing plans, manufacturing and assembly, and space shuttle scheduling.

A common thread through much of this work seems to be a need to analyze what are effectively *scheduling policies* as they might be implemented or revised for the organization under study. The key observation here is that while SLAM is basically a tool for simulating flows of discrete entities, there is often another component to the simulation that is programmed as an adjunct to the model, which implements a scheduling policy or algorithm and is, in fact, a central part of the subject of study. We believe that there is an important lesson to be learned from this in terms of the needs for next-generation tools. We will explore this more in the concluding section of this report.

ProTEM (Software Consultants Intl. Ltd.)

Release date: 1992 for Version 2.0 of the system.
Approximate cost: $1000
Number sold:
Source language: C or C++ (not clear)
Platforms supported: IBM compatible with Windows.

The ProTEM tool is advertised as a BPR tool; however, the graphical interface for model development is very much an object-oriented CPN modeling tool in that the graphical action flow model is a traditional Petri-net diagram, while the "tokens" are objects with attributes and behaviors that are triggered at transitions. Much of ProTEMs focus is on a methodology for moving from a business problem to a software model, using some of the more popular business analysis methodologies, especially the Zachman method. The Zachman Framework is a matrix model that helps analysts delineate all the different objects and processes involved in an enterprise. These are then modeled as real software objects in the model environment, drawing on a library of reusable objects as starting points.

While the system appears, based on descriptions provided, to be quite powerful in terms of what can be modeled using the tool, it seems clear that it still requires users to understand to a fairly high level both the terminology and structure of Petri-net models and modeling techniques, and also object-oriented coding practices. We did not have access to their library of business process objects, so we are unclear as to how well this library alleviates the problems sure to be faced by less experienced users, although Software Consultants makes the claim that this is so.

Workflow Analyzer (Meta Software Inc.)

Release date: 1993 Approximate cost: Design/IDEF ($4000) Design/CPN $24,000 Number sold: Design/CPN or Workflow Analyzer (unknown, but less than 50) Source language: ML and C Platforms supported: Macintosh, UNIX workstations

Metasoft (as they are generally known) has taken a somewhat novel approach to using CPN modeling for BPR. They have coupled their CPN simulation technology in a fairly seamless fashion to a popular IDEF0 graphical modeling tool they also developed called Design/IDEF. Design/IDEF is now their most popular product as a stand-alone tool on both Macintosh and PC/Windows platforms. It is described above in the section on Process description and modeling tools. Workflow Analyzer is actually a bundling of Design/IDEF with their simulation product, Design/CPN, and several other related tools and a consulting contract. Design/CPN runs on Macintosh and Sun Microsystems SPARC workstations.

One of the things that is interesting about this approach is that it enables much of the model development work to be distributed to line managers, as they have documented in the case of one of their clients, Shawmut Bank. These pieces are then collected, reviewed, and fitted together into one overall model which can then be used to generate a CPN simulation model almost entirely automatically. Of course, to actually run the simulation a large body of data must also be supplied about conditional branch probabilities, expected arrival rates, and so forth. This information is input into Microsoft Excel spreadsheets, which is the tool that Metasoft has adopted for all such parameter encoding. This too has advantages in terms of its steep learning curve.

Although it is clear that some successful applications have been developed using this tool suite, we have some hesitation about the process of going from IDEF models to simulations. IDEF is far less rigorous about how one describes processes than might be required for achieving accurate simulation of an organization. Some amount of training on proper modeling techniques and some massaging of the IDEF models before simulation generation is likely to be necessary, in our opinion, to make the models useful. However, we were unable to get first-hand experience in the use of this tool; therefore, we can only speculate about the kinds of problems that one might run into.

On the other hand, Design/CPN also supports its own graphical modeling language, that of CPNs, so there is a potential second opportunity to view and refine the simulation models based on this more powerful, if less perspicuous, formalism.

3.2.1.3 Object-Oriented Simulation Development Systems

Within the last few years, there has been a strong movement in the direction of object-oriented programming techniques, and the style of programming that this core technology promotes is especially appropriate for discrete-event simulation systems. As such, there has been a mass migration by all the major simulation vendors to products that are based on object systems. The simulation development environments in this category share a number of basic properties:

- There is a strong correspondence between objects in the world and software objects in the simulation environment.
- Reusable Object Libraries can be developed and maintained for different domains or classes of problems. These libraries tend to be marketed as add-ons to the core simulation systems.
- Simulation graphics and animation tool kits are naturally packaged as separate libraries.

MODSIM II/ SimObject (CACI)

Release date: 1987
Approximate cost: $25,000
Number sold:
Source language: C
Platforms supported: PC/Windows, OS/2, VAX/VMS, UNIX

MODSIM II is the current generation simulation language of CACI, one of the longest continuously operating simulation-development manufacturers. The language is a full-scale object-oriented programming language in the spirit of Modula II, Pascal, and ADA. The standard method (object behavior) model has been extended to include methods that behave "over simulated time" in a consistent fashion, enabling the full variety of discrete-event simulation behaviors to be modeled. Although originally designed to run on parallel processors as part of a DARPA project, it is now run primarily on single-processor platforms.

Due to its object-oriented basis, a number of add-on object libraries have been developed to make simulation development easier by starting from a higher-level foundation in terms of predefined object types and behaviors.

A full-featured graphical interface development object-library package is available called SIMGRAPHICS II. This provides much of the functionality for building sophisticated user interfaces and animated graphics.

SimObject is a baseline object library that includes primitives for developing simulation models graphically and, in turn, has several extensions or refinements that are prepackaged application frameworks for developing models in several classes of domains by refining the model that is there as a starting point. One is COMNET III, a simulator for communications networks. Another, currently under development, is SIMPROCESS III, an object-oriented version of the SIMPROCESS model described elsewhere in this report. A third is QUICKSIM, a prototype communications model built on SIMOBJECT. It is used for demonstrating some of the power of SimObject for graphical editing.

The SIMOBJECT editor provides a prebuilt extendible graphical user interface for building up models and simulating them. It is customizable to specific domains (like COMNET, SIMPROCESS) by adding new behaviors.

As a general purpose simulation development environment, MODSIM and SIMOBJECT represent a standard for present-day general-purpose commercial simulation development environments. However, for BPR-specific applications, there is still a fair amount of work that would have to be done to make this foundation into a tool that organizational designers could use themselves. SIMPROCESS III, when released, may move them a step closer to that goal.

IMDE (TASC)

Release date: 1993 Approximate cost: Number sold: Source language: Platforms supported: Sun SPARC with X-windows/OpenWindows

The Integrated Model Development Environment (IMDE) is a CASE tool that supports varying levels of users to work on the development of large simulation models. It was developed by The Analytical Sciences Corporation under an initiative of Armstrong Laboratory to alleviate difficulties associated with large-scale United States Air Force logistics modeling and simulation. It can best be thought of as a graphical development environment for MODSIM II simulations, although it can support the output of models into C++, and ADA is a desired third target language.

IMDE is basically a window-based environment for describing object classes and methods. In essence, it takes the MODSIM language and provides graphical support for implementing all of that language's programming constructs using graphical layout and form-filling operations. It supports different levels of (simultaneous) user/developers working on a large modeling project. It builds a model that is stored in an object database, supplied by Versant (a requirement for operating the system is the purchase of the Versant ODB).

While not requiring implementors to write lines of code directly in a language like MODSIM II, it does require the developers to be familiar with all of the functional programming constructs that such a language supports. That is, one must understand the object model and the different types of methods that one would use if programming with MODSIM II the "old-fashioned" way. However, by providing a palette of programming constructs and forms to fill in all the details, much of the low-level syntactic knowledge that one would need to master to use the text-based language is eliminated. A paper describing IMDE (Lloyd and Clark, 1993) states that simulations exceeding 15,000 lines of code have been produced without "writing a single line of code."

While IMDE may well improve the speed with which simulations can be developed and make maintenance easier, IMDE itself does not improve on the basic problem that MODSIM II has with respect to organizational process modeling. Namely, it is a general purpose development system which would need to be substantially tailored to work with organizational process concepts at an appropriate level of abstraction.

ARENA/SIMAN/CINEMA (Systems Modeling Corp.)

Release date: 1992
Approximate cost: $16,000 to $20,000; AST for manufacturing, an addl. $6,000
Number sold:
Source language: C and FORTRAN
Platforms supported: PC/Windows, OS/2, UNIX workstations

ARENA is an attempt to place an object-oriented paradigm on top of a non-object-based simulation language, SIMAN. (SIMAN was developed by the president of Systems Modeling Corporation, who previously was one of the main authors of the SLAM language.) Their purpose in doing this was to be able to support application-specific templates (ASTs), which are libraries consisting of modular building blocks for constructing hierarchical simulation models graphically from components with terms related to the manufacturing or business domain of interest. ARENA modules in the manufacturing area include notions of push and pull systems, queues, servers, and other features designed for modeling manufacturing systems, job shops, robotics systems, and materials-handling systems.

It was difficult to discern from the materials we had access to, but it appears that the ARENA graphical language for composing models in manufacturing is similar in power and scope to some of the other development tools in its class, such as SIMPROCESS, although their graphics may be much better.

G2 (Gensym, Inc.)

Release date: 1990 Approximate cost: $30,000 to $50,000, including training and support Number sold: Over 1000 in 23 countries. Source language: C Platforms supported: VAX, many UNIX platforms.

Gensym's primary market is large-scale, sophisticated, real-time control applications for complex manufacturing and engineering processes. Applications range from chemical and nuclear plant monitoring to space shuttle mission control to Biosphere II. This aspect of G2 is described briefly in the section on Real-Time Process Control systems. Here, we focus on several specific subsystems of the G2 environment, namely its general-purpose expert-system development tool aspects and its fully integrated simulation development system, which was originally designed for use as a test-bed for the real-time systems functionality.

Simulations are generally developed within G2 while real-time applications are under development. There are uniform interfaces to both the simulator and the real-time data collection facilities that make it possible for the simulation to be "swapped out" and the on-line application "swapped-in" at a moment's notice. Simulations in G2 are developed by laying out graphical models of plant flows (manufacturing control, such as chemical plant monitoring, being a typical application) and detailing the attributes and processes associated with objects along and in the path of those flows.

Because of its real-time focus, G2 simulations support models of both discrete and continuous processes, and work directly with fuzzy or uncertain values. Thus, the models can deal with business process flows either in the aggregate or as individual object flows with differentiating properties. G2 also includes a "Dynamic Scheduling Package" that allows users to develop or revise schedules when problems arise in on-line operations.

Several different BPR-oriented activities have or are going on with G2. Coopers and Lybrand have developed a proprietary BPR modeling facility using G2 that they use in their consulting practice. Gensym is now developing a BPR toolkit as an object library for use by third-party developers.

The development environment provided with G2 is an extremely sophisticated and easy-to-use expert systems development tool, complete with graphical object description and English-language-like rule description tools. This is all built on top of a uniform textual interface component that is *always* trying to complete every "sentence" for you, presenting you with all the possible next phrases, to which you can simply point or type. This makes model development rapid and relatively error-free.

Simkit (Intellicorp)

Release date: 1987
Approximate cost:
Number sold:
Source language: KEE (on LISP)
Platforms supported: UNIX workstations, PCs

No longer on the market, SimKit was an excellent early example of how multi-purpose simulation environments can be developed on top of an AI-based object and rule system application development environment. The original version of Stanford's virtual design team (VDT) workflow simulation was developed using SimKit. The current version of VDT is built on top of Kappa, Intellicorp's current C language version of their original KEE development environment.

3.2.2 Domain-Specific Simulation Development Tools

3.2.2.1 Business Process Simulations

In this category, we look at layered products on top of more general-purpose simulation modeling environments or simulation systems. Most tools in this category are either recent additions to general-purpose simulation environments or retoolings of more manufacturing-specific simulations.

SimFactory/SimProcess (CACI)

Release date: 1985
Approximate cost: $9,500 to $15,500
Number sold: approx. 1000
Source language: SIMSCRIPT II.5
Platforms supported: PC DOS and Windows, UNIX workstations

SimFactory and SimProcess are two variants of one system, designed for graphical model development of primarily manufacturing process simulations. The tools are geared to analysis of production capacity limitations, materials consumption, and rework history.

The graphical development environment of SimProcess/SimFactory is reasonably powerful and somewhat object-oriented, although the underlying mechanisms are not object based. Modeling is done by laying out objects for stations, queues, and transports, but the drivers are objects called "resources" that move through the system. That is, the objects that one might think of as being passed around in a factory or office, the object under construction, or the paperwork being processed are the objects for which one defines the flows, much as a sequence of steps (one per station) these objects must go through.

While this approach works reasonably well for most factory models, it is awkward for describing some kinds of flows, especially "pull" flows where things must be triggered by downstream needs. For example, it was not simple to build a model of an inventory mechanism that would issue reorders for parts when supplies dwindled. The CACI support line was not extremely helpful either, although through discussions with several CACI employees it became clear that one could model this behavior if one used some standard programming tricks (busy-wait loops) to check attributes of objects at the right time.

This tool could be quite powerful in the hands of a reasonably experienced model builder, but it requires some knowledge of simulation to use it effectively. On the whole, the model-building interface was not too difficult to learn to use, and the set of primitives one works with are closer to the right level for modeling a variety of business processes than most, although still at a finer level of detail than would be desirable.

CACI is in the process of producing a new version of SimProcess (SimProcess III) that will be built on top of MODSIM II and SimObject. This should prove to be an improvement both in terms of its true object orientation and the better graphics of the newer line of CACI modeling products.

CABRE ProSim/Witness (AT&T ISTEL)

Release date: 1986 (Witness)
Approximate cost: $24,000 to $50,000
Number sold: 1250
Source language: FORTRAN, C
Platforms supported: OS/2, VAX/VMS, UNIX workstations

Witness is a graphically animated manufacturing simulation system. It shows the flow of objects through a facility or process. While primarily for manufacturing process modeling, it has been combined with ProSim, the ISTEL suite of IDEF modeling graphics tools, to support business process simulation. Much like MetaSoft's use of IDEF0 as a modeling environment for building simulations, CABRE uses primarily IDEF3, which promotes more detailed models than IDEF0 by including such things as explicit conditionals, GOTOs, and parallelization. Object-state transition diagrams are also used to model the changes that objects can go through as they are simulated.

ProSim is a combination of the ProCAP (described earlier) IDEF3 description tool and an "expert system" that seeks to at once hide the "programming tricks" that a developer might need to use to get realistic behavior and enforce enough detail in the model developed to make it simulatable. Unfortunately, the paper describing ProSim that we were given suggests that one still needs a *thorough* understanding of IDEF3 before beginning to work with this modeling tool, and the WITNESS simulation tool seems like it may not be the best fit in many business process modeling projects as the simulation engine, given its strong manufacturing focus.

Extend+BPR (Imagine That)

Release date: 1993
Approximate cost: $990
Number sold:
Source language: ModL (a modeling language built on top of C)
Platforms supported: Macintosh

Extend was covered earlier under the Numerically Based simulator heading. The object-oriented character of the system allows it to be easily extended; this led one of Imagine That's clients to develop a BPR modeling library package for Extend.

The BPR modeling library for Extend uses the discrete-event modeling library and builds on top of that a few basic process blocks for more business-oriented modeling, such as queues, transactions, and decision processes. It includes some support for activity-based costing. However, it is not the tool for complex modeling jobs.

ReThink/G2 (Gensym, Inc.)

Release date: under development
Approximate cost: $30,000 to $50,000 for G2, including training and support
Number sold:
Source language: G2 Object language
Platforms supported: VAX, many UNIX platforms

ReThink is a business process flow modeling simulation, still under development, built on top of G2's simulator and modeling environment. While the G2 environment makes for a powerful starting point, the range of constructs that have been developed for the business process application area is limited as yet.

3.2.2.2 Workflow Modeling

These are tools that specifically model inter-personal communications by different modalities, meetings, and so forth, and workflow in office environments as well as the usual task performance delays.

Workflow Simulation Model (Arthur D. Little, Inc.)

Release date: 1991
Approximate cost: Not sold as a product
Number sold: Used in several dozen studies
Source language: FORTRAN and SLAM
Platforms supported: PC/Windows

Arthur D. Little, Inc., developed an application of SLAM for studying electronic and paper information in workflow processes that they have used in a number of consulting studies. This system shows both what can be done and, to some degree, the difficulty of doing complicated process modeling with Pritsker's SLAM system. The model they developed implements a set of probabilistic flows of paper processing tasks through a system. For example, some external FORTRAN routines had to be built to complete the environment, and this required a detailed knowledge of the data structures of the FORTRAN-based SLAM environment.

The model itself was designed to look at such tasks as the conversion of paper documents to microfilm or electronic form, and the subsequent recall and use of those documents. Other kinds of processes include bank handling of checks or money orders, loan processing, and other high-volume transactional processes. Issues to be modeled included such things as activity cost, time to completion, staffing and shift requirements, numbers of handoffs, and other measures of inefficiency.

Many tables of inputs to be processed over time, and probability tables indicating how they may be diverted at branches in the model, are carefully set up before each run. The output of the models is primarily textual data files that can be loaded into spreadsheets for further analysis, beyond the few types of graphs that can be generated automatically using SLAMSYSTEM.

VDT (Stanford University, Center for Integrated Facility Engineering)

Release date: none -- a laboratory research prototype Approximate cost: N.A. Number sold: N.A. Source language: ProKappa -- Previously KEE and SimKit (Intellicorp) Platforms supported: UNIX workstation

Built originally using Intellicorp's KEE and SimKit technologies, and more recently re-engineered on top of ProKappa, VDT (for Virtual Design Team) is a simulation model for large-scale routine engineering design tasks. That is, it is an attempt to model a team of engineers in the process of doing a design for a large engineering project. It is described in several papers, including *The Virtual Design Team* (Levitt et al. 1993).

The VDT model is based on Jay Galbraith's (1977) information processing framework and micro-contingency theory. It combines some of the more standard process flow model notions with a more explicit model of the demands on individuals in the work process, including interruptions for phone calls, meetings, and other information-processing capacity issues. It models the differences in different communications channels explicitly, such as phone vs. email vs. personal communication.

VDT models agents as having to allocate their attention to tasks. It explicitly models authority relationships and can use that in determining how an agent prioritizes what task to do next based on where the task came from. It assumes a mix of task selection strategies are used by each agent (sometimes do the next thing in the queue, sometimes respond immediately to a phone interruption, sometimes pick the most important thing to do next. . .).

The model was validated by reproducing the overall timing behavior of a 3-year petroleum refinery design project having a total design/construction cost of about $130 million, a budget duration of 20 months, and a peak staff of 120 managers, engineers, designers, and support staff located in two offices. The modelers were able to predict that centralized decision-making led to longer task duration than decentralized decision-making. The modelers also showed that voice-mail improved the overall performance.

Although the range of domain models VDT was intended for is limited (to organizations doing routine design), many of the ideas that are implemented in this simulation model apply to a wide range of business organization activities. We see this work as an important step in the direction of better, more detailed models of white-collar business activity.

3.2.2.3 Manufacturing Simulation Environments

Many of the simulation tools in the commercial market were developed to support the investigation of manufacturing processes and still see that as one of their primary markets; thus, some tools are strongly dedicated to that kind of application. This section examines those systems.

3.2.2.3.1 General Manufacturing Simulation Environments

The tools in this category are simulations that are geared directly to manufacturing modeling, although not industry-specific. They tend to have powerful graphics to show what happens on a factory floor as the simulation proceeds. SIMAN and SimFactory, described elsewhere, are other packages that have been used extensively for this kind of application, although they have more general applicability.

AutoMod, AutoSched (Auto Simulations)

Release date: 1993 (V6.0)
Approximate cost: $25,000
Number sold:
Source language: C, C++
Platforms supported: Sun SPARC, other UNIX workstations, PC/Windows

AutoMod is an engineering/simulation tool for manufacturing, materials handling, storage, and distribution system design and evaluation. It includes a well-integrated three-dimensional graphics system for detailed, precise animation of simulations of factory processes. Models can be constructed using spreadsheets or an English-like language. Manufacturing systems are described by building data tables for resources (machines, people, tools), processing steps (part routing), and production quantities (orders). AutoMod can import CAD layouts. This is a high-end tool for the manufacturing market.

AutoSched is a finite capacity planning and scheduling tool. It is built on top of AutoMod. It allows users to insert rules to model how machines and personnel select tasks, and to simulate the implementation of those rules. The end result is the discovery of more effective scheduling rules that can improve performance. It is an example of the kind of modeling activity that we expect to see more of in the future, as BPR simulations get into the issues of individually scheduled work within an organization.

Factor/AIM (Pritsker Inc.)

Release date: 1991
Approximate cost:
Number sold:
Source language: C
Platforms supported: OS/2

AIM is a manufacturing simulation system in which users build models by locating machines, operators, and material-handling equipment directly on-screen and filling in the details of the model in pop-up forms. It accommodates the importing of data from spreadsheets and databases. It includes constructs for modeling common manufacturing elements such as machines, operators, conveyors, AGVs, fixtures, robots, buffers, shifts, breakdowns, preventive maintenance, work-in-progress areas, parts, orders, and process plans. Equipment can be added or removed to show affect on system throughput. AIM models machine breakdown rates, management of inventory levels, and lot sizing. Order-sequencing can be modeled with built-in rules, and conversion to just-in-time processing can be modeled also.

Similar to SimProcess/SimFactory, process plans are defined from the point of view of the part moving through the factory. That is, one defines the sequence of activities that happen to the components moving through the plant, using a form-based interface.

Factor/AIM is part of an integrated Finite Capacity Management System; the other two components are a Schedule Development module and a Schedule Management module. Both of these are designed primarily for operations support.

3.2.2.3.2 Industry-Specific Manufacturing Simulation Systems

This subsection presents examples of simulations used in manufacturing process planning for two very different industrial settings: chemical plant design and high-precision robotic assembly tasks.

Mirror Model (ChemShare Corp.)

Release date:
Approximate cost: $50,000 to $500,000 (including on-site analysis)
Number sold:
Source language: FORTRAN
Platforms supported: UNIX workstations; PC/DOS

This product is targeted specifically for continuous-process industries such as petrochemical, chemical, or gas-processing plants. Because the mathematics of continuous chemical processes are well understood and can be rigorously specified, a very accurate "mirror-like" simulation model of an entire plant can be constructed with this tool. It enables the user to precisely determine the real-time, ongoing state of a process even though many of the process measurement inputs to the model contain statistical and/or *gross* error. The model can be used to set optimum setpoints for the process, balance materials needed, locate faulty measurement instruments, and determine equipment efficiency. Operated off-line, the model can be used to try out different operation scenarios to optimize the process.

Virtual NC (Deneb Robotics Inc.)

Release date: 1993
Approximate cost:
Number sold:
Source language:
Platforms supported: HP, Intergraph, Silicon Graphics, Sun; PC

Virtual NC is a high-precision graphical animation simulator for machine-process robots. The system emulates the complete functionality of both a robotic arm and its NC (numeric control) controller. The user is presented with high-resolution perspective three-dimensional graphical displays of the robotic arm in motion. The software includes capabilities which allow the user to emulate any NC controller and to verify its software. From simulation runs, the user can generate tailored reports listing such problems as cycle errors, collisions, axis over-travels, and excessive depth of cut.

PROsim (Viewlogic Systems Inc.)

Release date: 1993
Approximate cost: $6000
Number sold:
Source language:
Platforms supported: DOS and Windows

PROsim is a timing simulator primarily for electronic circuit design analysis targeted to entry-level systems designers. PROvhdl is a digital simulator layered product on top of PROsim. It is a numerically based continuous/discrete process simulator.

3.3 Process Control and Process Management

While we are focused primarily on tools to aid in the design or redesign of business processes, it is useful to look at some of the tools in the marketplace that address the **on-line** needs of organizations to manage data and processes. This is particularly true since many of these tools provide, by necessity, modeling capabilities by which tailored application solutions are constructed, and these models reflect precisely the processes the designers intend the organization or manufacturing system to follow.

3.3.1 Workflow Applications Development Systems

The four packages discussed below all support the design, development, implementation, and management of **workflow applications**. In general, each package consists of several distinct software tools, some of which may be sold separately. Typically, there will be a tool for each of the following activities:

- analyzing and designing the workflows;
- building the actual workflow applications;
- managing the flows of work, typically on a file-server;
- running the applications on each individual's desktop, typically on a PC; and
- supporting the collection and analysis of data on the progress of work.

Some packages include a tool to build user interfaces for the applications; some support general-purpose tools for that, such as Visual Basic or PowerBuilder; others integrate such capabilities into the application-builder. The packages vary considerably in the extent to which these distinct components are integrated and, thus, the extent to which one component or tool can automatically incorporate the results of using another tool in that package.

We have included a fairly extensive discussion on workflow in general for three primary reasons:

- it is an important emerging class of software to support organizations that has often been coupled with the idea of business process re-engineering;
- it is an area in which we believe the Human Resources Directorate of Armstrong Laboratory has some interest; and
- it is an area where we believe important advances can be made by investing research and development resources, a topic we address in Section 5 of this report.

Workflow software is for work that can be broken up into steps; generally, each individual step is done by one person at a time, although the entire workflow usually goes through a number of steps and people. Processing an insurance claim provides a classic example: it might have many steps, done by many people, but each step is done by only one person and in a clear sequence (with parallelism allowed). It is

important to distinguish it from **groupware**, software that people use when they are **working together**, collaboratively and interactively, often on relatively ill-structured problems. A useful heuristic is that groupware supports what people might have a meeting for. Virtual meetings of various types are the most common form of groupware, but there are others as well.

Workflow is often linked with imaging software (indeed, it grew out of imaging and needs for image management). Its original application was focused on routing "documents" to which some actions had to be applied; without some such software, large image storage systems are much less useful.

It generally provides (graphical) tools for process modeling or "building process maps," as they say. These then serve as the basis on which specific applications and data entry screens are built. It may have hooks to generate database structures from detailed descriptions (to the level of specific information requirements) of the steps. It generally provides support for building "applications," which means forms to be filled out in each step, queries to be sent off to databases, and routing rules. Some systems support rather complex conditional routing rules, while others only allow for rigid routing. It generally provides for management information and control. (For example, managers can monitor how long it takes people to do steps, how long queues are, where bottlenecks are, etc.).

Workflow gets its leverage in several ways:

- There is no longer a unique original physical file, to which almost all things have to be done; this makes some parallelism possible.
- Time spent in interoffice mail is eliminated, and electronic work shows up right on the computer screen of the person who is supposed to do it next.
- A good audit trail of who has done what when can be developed.
- Tracking and obtaining status information on work-in-progress is greatly facilitated.
- A range of other communication overheads are also reduced, such as those associated with passing on work from one person to the next.
- Workers are enabled (indeed, forced) to focus on the value-adding components of their work, and managers are given tighter monitoring and control capabilities.
- Interaction with the database is (at least somewhat) automated, thus saving time.
- Errors (and thus very expensive rework) are reduced by having mandatory fields with range checks on automated forms (which makes it impossible for the work to move on with an inadmissible or missing value) and automated lookups.
- Sometimes it is possible to redesign the process being automated, saving time and money.

Workflow is by no means the same as business process redesign. However, it makes sense to think through the processes and perhaps redesign them before implementing workflow. The software itself generally offers very little support for this (e.g., no real simulation capabilities are currently available). The way to do BPR in the context of workflow is to hire consultants. The big six accounting firms and other management consulting organizations do these sorts of engagements: help to identify suitable processes (if that hasn't been done by the client); make a "process map"; design a workflow implementation, with some degree of BPR involved; help to select vendors and products; and help to implement the workflow solution.

These products run on PC clients, typically with a UNIX server. You make a "process map" of the steps or activities; you "drill down" and specify the information needs for each step in detail; you use an API (Application Programs Interface) to build applications that will present partially filled out forms to users; and you use a package (typically Visual Basic) to design the interfaces for the forms. Some of them use email as a transport medium; others use Lotus Notes. Some use databases to keep track of workflow events, while others use the available mechanisms in Lotus Notes.

Some of the products are coupled with a particular image management system; others can integrate with many. Some support complex, conditional routing rules (e.g., "if the claim is greater than $1000, send this claim form to the supervisor in category X with the shortest queue; else send it to the payment clerk"); others support only a clear path. All link with some database; some claim to work smoothly with any SQL database. They vary widely in the extent to which they can integrate with other applications on the server, such as expert systems. They work well over LANs; more geographically distributed workflows, using Wide-Area Networks (WANs), are at the cutting edge of existing technology.

Many workflow users have legacy COBOL systems on mainframes, and understanding integration with legacy systems is an important capability that consultants must offer. Often, developing suitable integration with legacy systems consumes the largest amount of the work in building and implementing a workflow system. Some consultants try to leverage their BPR capabilities to do this work; others are more focused on workflow *per se*.

Workflow technology is developing fairly rapidly, and it seems likely that there will be substantial advances in the field over the next six to twelve months. Indeed, vendors who were into this technology early tend to use less advanced approaches, and it is quite possible that new leaders will emerge in the medium term.

Action WorkFlow (Action Technologies, Inc.)

Release date: 1993
Approximate cost: Analyst, $500; Application Builder, $1500; Manager, $6500 to $8000 per server and up to 10 users; $500 to $750 per user beyond 10.
Number sold:
Source language:
Platforms supported: Windows 3.X and Lotus Notes on IBM-compatible PCs for applications on client; Workflow Manager runs under OS/2 or windows NT with Lotus Notes or SQL Server, on the server, which must be an IBM-compatible PC

The Action Workflow System consists of four key components:

- Action Workflow Analyst, which runs under Windows 3.X in stand-alone mode, is used to build a "process map" of the business process. It consists of a set of graphical tools that support drawing Action workflow diagrams, which consist of "Action workflow loops," discussed below.
- Action Workflow Application Builder, which makes it possible to build applications based on process maps developed with the Analyst, runs under Widows 3.1 or later on IBM-compatible PCs, either stand-alone or in networked mode.
- Action Workflow Manager runs on the server and coordinates the workflow and the applications running on individuals' machines.
- Action Workflow Service Application Programming Interface, which makes it possible to link other applications, including legacy systems, into the workflows.

The critical feature that distinguishes the Action approach to workflow from others is its grounding in the theories of Terry Winograd and Fernando Flores (1986) that view language as based on commitments and communication as a form of human social action (building on the work of Martin Heidegger). The approach takes the essential atomic event within a workflow to be an interaction between two people; a large and complex workflow, then, is built out of many such dyadic interactions.

In each such interaction, one party is the "customer," the one for whom something is being done; the other party is the "performer." Each interaction has four phases: in the "preparation phase" a proposal is made of work to be done, which is usually a request by the customer but can also be an offer by the performer. In the "negotiation phase" the customer and the performer agree on the "conditions of satisfaction" for the work, which is typically expressed by specifying the time by which something will be done. In the "performance phase" the work is done, and in the "acceptance phase" the work is assessed against the conditions of satisfaction and either accepted or not. Each phase consumes some time.

An interaction as described above is the "Action Workflow loop." The Workflow Analyst supplies automated support for drawing such loops and for providing the details that can go into them (which will not be discussed at any length here but

include distinguishing primary and secondary workflows and parallel and serial workflows, identifying the twelve basic acts, and distinguishing among normal, conditional, and exception links between workflow loops). Workflows can be hierarchically embedded. Developing a process map does not require a computer programmer.

Once the process map has been developed, a programmer must be brought in to use the Application Builder to develop the actual applications and associated work routings. At this stage, the information requirements and procedures for each step must be specified in detail, so that they can be automated. Code must be written to link to other applications and to access data. Roles and names of individuals in those roles are also specified. (All workflow systems have similar mechanisms to control routing.) The data definitions developed at this stage are used to drive the database tables in the Workflow Manager, which keeps the workflow going.

The major advantages and disadvantages of the Action approach stem from its basis in dyadic interactions between people. For work processes that are relatively heavy on such interactions and light on information processing steps *per se* (for example, those involving multiple handoffs for review and approval, with a web of completion dates), Action provides a powerful approach. It forces the analyst to focus on the human interactions and to make them explicit. In addition, the integration of Action's products with the Lotus Notes platform provides a strong base for implementation.

For example, at the Young and Rubicam advertising agency, Action Workflow was used to provide automated support for the process by which specific advertising jobs were initiated and approved for one large account. The process was already well-documented but was paper-based and required a considerable number of personal communications (face-to-face or by telephone) to track progress and keep it moving smoothly. Since there were a number of people involved in many of the steps, the opportunities for breaks in the process were considerable. Also, since there was relatively high turnover in the agency, the paper-based system made it difficult for new people to come into the middle of a process and know where they stood relative to it. The forms used were automated and, most important, the routings of those forms and associated information was also automated. This made it possible for each person to have a common representation of the process and its current status, and for account coordinators to obtain an overview of status on demand. The implementation was on top of Lotus Notes, and took advantage of many of the features of Notes to manage and track routings. (This analysis is based on Marshak 1993.)

The primary disadvantage of the Action approach is that it is not particularly well-suited to processes that are **information-intensive** as opposed to **interaction-intensive**. In fact, the aspects of the process dealing with the information content and transformations at each step must be recorded as free-form text notations to the Action workflow map made with the Analyst. This puts much of the onus on the application builder to discover the details relevant to the information aspects of the process, and he or she typically will have much less contact with users during the application-building stage than during the process mapping stage. For example, Action's approach is not as well-suited for a process such as processing an insurance

application or claim, or automating the steps in purchasing from submitting a purchase requisition to receiving what was ordered, delivering it to the appropriate employee, paying for it, and maintaining an inventory entry.

ImageWorks/FlowPATH (Bull Information Systems)

Release date: 1993 Number sold: 15 (none in the U.S.) Source language: Platforms supported: Bull DPX/20 and DPX/2 Unix servers; PC clients running Window

FlowPATH has just begun to be available in the United States, about a year after its release in Europe, where Bull is headquartered. Although it interfaces well with Bull's IMAGEWorks image management system, it was developed independently of it and is being used independently by some customers. It is a full workflow system; in general, it is strong on its internals and less strong on the externals, such as user interfaces.

Workflows are analyzed in terms of procedures, which are composed of activities and subactivities. Activities and subactivities can be fully automated (such as a calculation, database lookup, and the like), fully manual (such as a person entering some information into a field), or mixed (such as a person making a credit assessment based on automated data lookups and calculations). Activities or subactivities must be scripted, which involves specifying the data they use, the actions that can be taken on those data, the time an activity takes, and the interval necessary between different instances of the same activity. Activities are executed by people via the assignment of people to roles. A specific instance of a procedure is a workcase, and a specific instance of an activity is a task. In building a workflow, time-outs, automated alerts, and delays in forwarding work to the next person can all be built in.

On the desktop clients, work is presented in queues which list the activity and some associated information. A user initiates an activity by clicking on it in the queue, which can also summon the relevant application. Users can also attach annotations to workcases, and read annotations attached by those who have worked on it previously.

Management information is collected on all workcases and can be accessed on-line and analyzed after the fact by the workflow coordinator. The coordinator also has the capability to reroute work dynamically, based on either analyzes of work in progress or on automated alerts about bottlenecks or delays. A workflow administrator is also defined who has a different set of tools to support implementation, maintenance, and administration of an installed FlowPATH system.

The MOBILE modeling tool is used to build graphical models of procedures, following the conventions of the information control network (ICN) approach to modeling flow of control in business processes. Some of FlowPATH's special features come from its reliance on ICN. ICN explicitly represents activities, their precedence relations, the information they use, the repositories where that information is found, and parallel activities. The graphing convention enforces control on parallel activities by requiring that both "and's" and "or's" be closed; this is reflected in the underlying ICN and ensures that there can be no dangling activities, which provides one form of integrity for the model.

FlowPATH uses data vectors to specify the data used by activities; the initial specification of these is provided by the MOBILE tool, and they must be further specified, in general, when building full workflow applications. Data in such vectors can be either actual data or references to where data is stored, including in a remote database. FlowPATH also has a simulation component which uses the underlying ICN model to simulate the workflow. To run a simulation, the user must specify several simulation parameters, such as the rate at which workcases are initiated, the total number in the simulation, and whether they come singly or in bundles. Although the simulation is fairly crude and only provides for measures of queue length and throughput at each node in the graph, it can be used to identify bottlenecks.

An additional virtue of FlowPATH is its modular architecture, which makes it possible to substitute for many of its pre-existing modules. For example, it has a load balancing function which routes work to the person in the required role who has the shortest work queue. But a more complex routing algorithm could be used which considered priorities of work, gave a preference ordering of the people within a role, and the like. Due to the modular design, it would not be necessary to modify code in multiple places to make such a change.

FlowPATH is strong in its ability to associate data (of various media types, including images and audio) with activities and provides a useful formalism for modeling information-rich workflows in its implementation of the ICN approach. It is weaker in the post-hoc analysis tools that it provides and in some of its operating constraints, (such as size of diagrams which can be drawn with MOBILE and some of its network characteristics).

ProcessIT (NCR)

Release date: 1993
Approximate cost: $1995 for development kit; $495 for Process Activity Manager, workflow engine that runs on server to manage the workflow; $995 per seat for applications, two $95-per-seat optional add-ons
Number sold:
Source language:
Platforms supported: NCR Unix servers; IBM-compatible PCs running Windows 3.1 or greater for clients

ProcessIT is a full workflow system oriented toward transaction processing applications; it supports NCR's image management system but is also being used at a number of sites without any image capabilities, strictly to support workflow. Process Maps are used to diagram the workflow, in a format similar to flowcharts, using the graphical Map Builder. (These maps permit embedding and naming of "submaps," which can then be readily reused in other process maps.) The maps specify routings and routing rules. Each processing step in such a map is an activity, has data associated with it, and can have an application associated with it as well. This makes it possible for an application to be evoked automatically when the relevant activity is initiated. (Thus, for example, a spreadsheet might be activated for a certain activity performed as part of the mortgage application and approval process.)

Individuals are assigned roles, and activities are also assigned to roles. Specific instances of work are termed "couriers" (capturing the idea that they move around, taking work-in-progress from place to place), and facilities in the Administrator (part of the development kit) make it possible to track couriers and summarize work by activities, users, and processes. Dates can also be used to index and track work and work-in-progress. ProcessIT has a broad range of application-programmer interfaces and can interface to any code that can interface to C.

ProcessIT has several useful features for dealing with routings. It permits dynamic rerouting, which means that a designated administrator can redesign a process that is running. It also allows for guaranteed routing, which prevents specified routings from being overridden while they still have work in progress. (New couriers on that route, however, would follow the new routing.) It supports parallel routing (for example, for multiple simultaneous reviews) and has a capability that can require that parallel couriers be rejoined before the work can move on.

The facilities ProcessIT provides for building the data flows and access paths are not as strong as some of its other features, although it does provide great flexibility for this through its APIs. It is unusual among workflow products in that it has a simulator as part of its suite of tools, which makes it possible to simulate a designed workflow before implementing it. (It does this using simulated couriers.) To do so, the user must specify various parameters of the activities in that workflow (such as processing times and error or rework rates); the results can be analyzed with the same tools used to analyze operational workflows. Note that the simulator does not simulate what occurs *within* each activity, and problems with data access or mismatch

of applications evoked and data provided would not be discovered in the simulation. Nevertheless, it does provide the capability to assess various workflows which use the same activities and simulation parameters, which can be useful in attempting to tune a workflow design.

Visual WorkFlo (FileNet Corp.)

Release date: 1993
Approximate cost: $11,500 developer's kit; $1800 per desktop for applications (which includes ability to run imaging applications)
Number sold:
Source language: C
Platforms supported: Servers, IBM RS6000, and HP3000 and 9000; clients, IBM-compatible PCs

FileNet Corp. is a major player in the imaging business and recently released object-oriented software for workflow intended to integrate with its imaging systems and, potentially, to appeal to a broader audience as well. Thus far, virtually all users of Visual WorkFlo have been existing customers of FileNet's imaging products. The **WorkFlo System Development Kit** provides a range of graphical tools and a library of job functions for building workflow applications. Each step in a business process is treated as a software object, which makes it possible for steps to be readily combined in different ways or modified, if necessary, as a business process changes.

WorkFlo Performers run on the desktop and are how the work gets done; they can include work steps that must be performed by people (often with associated images of documents), by programs or scripts written in any of a variety of languages, or by existing applications. The **WorkFlo Conductor** runs on the server and keeps track of Performers and the progress of work; it supports system administration and management reporting. There is also a component that makes it possible to run workflow applications over a wide-area network.

The focus of FileNet's approach is on adding value to their imaging systems by making it possible to integrate images more effectively into the ongoing work of an organization. FileNet applications can be integrated with Lotus Notes but do not require Notes as a transport medium. Their tools are not strongly oriented toward the analysis stage and are not strongly grounded in a particular point of view (in contrast to Action Technologies, for example).

3.3.2 Real-Time Process Control Systems

Tools in this category are all geared to on-line monitoring and automated control of manufacturing processes. These tools have varying levels of sophistication in the complexity of the processes they can monitor and predict problems in, and in the sophistication of the background models and graphical displays that can be developed.

RT-DAS V3.0 (Talton/Louley Engineering)

Release date: 1993
Approximate cost: $1000 to $3000
Number sold:
Source language:
Platforms supported: PC/MS-DOS, LANs

RT-DAS is an open-architecture process engineering system for real-time data acquisition. It provides continuous or batch processing, laboratory analysis, distributed control, and statistical process control. It designs/develops process control strategies. Its applications include Supervisory Control, Direct Digital Control, Management Information, Data Acquisition, Laboratory Experiments, Temperature/ Pressure/ Flow Systems, Pilot Plant Production, Test and Measurement, Boiler/ Furnace Control, Environmental/ Sterilization Chambers and (Continuous) Simulation/Modeling Systems. It does **not** include a simulation tool.

Typical industries for RT-DAS include oil and gas, utilities, petro-chemical, food, pulp and paper, pharmaceuticals, aerospace, pipeline, and waste management.

Universal Control System (Taylor Industrial Software)

Release date: Most components had Windows versions in 1993
Approximate cost:
Number sold:
Source language:
Platforms supported: MS-DOS, Windows

A suite of tools for Manufacturing Control in Computer-Aided Manufacturing (CAM), it includes:

- SecureWORX for programmable device support;
- I&CS, an Instrumentation and Control System;
- OBJEX, a function block controller that provides a graphical O-O environment for PC-based direct and supervisory control applications;
- FloSOLV, a concurrent flowchart controller for planning floor control;
- Process Window MMI, for visual representation of manufacturing processes and machines with real-time animation; and

- MRS (Manufacturing Resource Scheduler), a decision support tool for batch-oriented manufacturing environments which provides more than 60 predefined reports.

G2 (Gensym Inc.)

Release date: 1990
Approximate cost: $30,000 to $50,000 including training and support
Number sold: Over 1000 in 23 countries.
Source language: C
Platforms supported: VAX, many UNIX platforms

Gensym's primary market is large-scale, sophisticated, real-time monitoring applications for complex manufacturing and engineering processes. Applications range from chemical and nuclear plant monitoring to space shuttle mission control to Biosphere II. As with other products in this category, G2 combines mechanisms for describing rules that should "fire" when critical conditions occur in the process being monitored, and also does modeling that will allow one to predict conditions given certain kinds of inputs. G2 as a whole is a powerful application development tool that includes database access, real-time data collection and monitoring functions, a general-purpose object-oriented (tick-based) simulation environment, and a full object and rule language support. These general development tools were discussed earlier.

3.3.3 Manufacturing Data and Process Management

This first category of tools largely predates the recent wave of interest in designing business processes *per se*. These are somewhat more traditional, yet highly detailed tools for developing sophisticated management information systems for manufacturing companies. They are not simulation tools in any real sense; thus, they are not our primary focus in this study.

BPCS-Manufacturing Data Management (System Software Associates)

Release date: 1993
Approximate cost: $5000 and up
Number Sold: 2000
Source language: RPG III
Platforms supported: IBM AS/400

This system keeps product design and process engineering data (bill of materials) on-line for use by all manufacturing planning, production activity control, and costing functions. Provides multi-plant and alternative manufacturing methods support. It is customizable in a number of ways.

DMACS (ONLINE Software Labs, Inc.)

Release Date: 1983
Approximate Cost:
Number Sold: 55
Source Language: COBOL
Platforms supported: UNIX-based systems

DMACS stands for Distribution, Manufacturing, Accounting, Costing, and Simulation. It is a UNIX-based client/server business planning and control system for distribution and manufacturing organizations. Through a series of software modules, it provides forecasting and inventory management reports, sales analysis reports, customer order management, master scheduling functions, purchasing, work in progress, engineering, costing, sales commissions, customer rebates, and a range of accounting functions.

Adaptable Manufacturing System (Adaptable Business Systems)

Release date: 1993 Approximate cost: $9,800 (includes source code) Number sold: Source language: BASIC Platforms supported: UNIX platforms

AMS is an integrated management information system for manufacturers and job shops. It includes modules for finance and accounting, marketing and sales, product and process engineering, purchasing and inventory management, production control and capacity planning, and system utilities.

The product and process engineering system coordinates product master entry and reporting, bills of material entry and reporting, bills of material where-used reporting, work centers, machines and labor grades, and process and routings entry.

Mercury ISPA V3.0 (AI Technologies Inc.)

Release date: 1993 Approximate cost: $35,000 to $70,000 Number sold: 10 Source language: Mercury KBE (LISP) Platforms supported: DEC VAX/VMS, UNIX platforms

Mercury ISPA is an intelligent statistical process analysis package that generates models for process control, quality control, process development, and simulation. It includes an object-oriented process knowledge-base and statistical master to guide creation of models. ISPA detects model degeneracy and tests for model fitness. Mercury KBE is an AI-based applications development environment.

3.4 Miscellaneous Products

3.4.1 General Object-Oriented Application Development Environments

The tools in this category are general-purpose intelligent business application development environments. They all have their origins in AI technologies and all include sophisticated object-oriented modeling components, rule editors and interpreters, graphical presentation facilities, and database access mechanisms for a variety of relational and/or object-oriented databases. As such, they are designed for rapid prototyping and are all reasonably well suited to the development of object-oriented simulations, although for the most part they do not provide special facilities for that kind of use (with the exception of G2).

ART Enterprise (Inference Corp.)

Release date: 1993
Approximate cost: $7,000 (Mac, PC), $10,000 (UNIX platforms)
Number sold:
Source language: C
Platforms supported: Macintosh, PC/Windows, OS/2, UNIX platforms, IBM mainframes

ART enterprise combines the expert systems-building capabilities of Inference's original ART environment with additional GUI (graphical user interface) and database access tools to position Inference for more mainstream business applications development. It is a development tool for rapid prototyping and deployment of object-oriented programming systems that can also take advantage of rule-based inference processes. It is based on an event-driven client/server open architecture that allows access to unstructured information and business-rule-driven processing. It supports "what-if" kinds of analyses, and has GUIs for multimedia kinds of presentations.

Level 5 Object (Information Builders, Inc.)

Release date: 1993
Approximate cost: $995
Number sold: 5,000
Source language: C
Platforms supported: PC/Windows, VAX/VMS, IBM mainframes

The "low cost" version of an AI-based applications development environment, Level 5 claims to support many of the same features you would expect to see in the higher-priced environments (although it's hard to tell how it compares quality-wise without further investigation). Level 5 supports all the standard kinds of rule-based reasoning that are common in these environments; it supports object-oriented programming, including support for object-oriented databases. It supports access to over 70 different database server products on local and remote platforms (including many of the popular PC-based systems). It has a powerful graphical user interface GUI system

that gives full access to Microsoft Windows. It supports use as a server in a client-server arrangement.

Kappa (Intellicorp)

Release date: 1992
Approximate cost: $26,000
Number sold:
Source language: C
Platforms supported: UNIX workstations, X-windows, PC/Windows

Kappa is Intellicorp's current generation AI-based applications development environment. It includes a full interactive C development environment and API, a GUI development system, dynamic browsers and debuggers, a dynamic object system and object query language, and access to traditional databases, CASE tools, and other legacy systems. It is driven by the ProTalk language (Kappa used to be called ProKappa), which is a hybrid of procedural fourth-generation languages and object-based languages. It contains an object query and pattern-matching engine (like an "SQL for objects"). Applications of Kappa can be ported from UNIX to PC/Windows environments in a few hours or less.

The Object Management Workbench (OMW) implements the Martin/Odell Object-Oriented Information Engineering Methodology. It is a visual development environment that provides full life-cycle support from analysis and design through delivery. It supports CASE modeling and executable business models with high-level expression of business policies. It is designed for rapid iterative development of business applications. It contains an event flow diagrammer that supports event-based triggers driven by business rules.

The current version of the VDT simulation (see workflow simulation section) done at Stanford University was developed using Kappa.

G2 (Gensym Inc.)

(See description under Real-Time Control Systems.) Although not directed at this market, in price or emphasis, G2 is a very powerful general-purpose object-oriented application development environment in its own right.

3.4.2 Probability Propagation Tools

The two tools in this category are both add-in's for spreadsheets that enable a user to statistically quantify the uncertainty in a forecast made by a spreadsheet model as an aid to risk analysis and decision-making. Four steps are involved in their use. First, the user must specify which spreadsheet cell is the forecast output variable and which spreadsheet cells are the input variables that involve uncertainty. Second, for each uncertain input variable, the user must specify the distribution of values expected, by choosing from among several types of statistical distributions and specifying required parameters (e.g., mean and standard deviation, or mean, minimum, and maximum). Both tools permit the user to specify correlations among input variables. Third, the user specifies the mode of random number sampling (Monte Carlo or Latin Hypercube) and the number of iterations to be run (the sample size). Finally, the user runs the simulation and the program plots a histogram of the distribution of values of the output variable and provides other tools for analyzing and understanding the results. The value of these products is in answering questions such as, *How likely are we to be under budget for this month?* or *What is the probability of finding more than 20,000 barrels of oil at this site?* Single-point estimates from a spreadsheet are often unrealistic in not capturing the uncertainty of a forecast resulting from variability in one or more input variables. These products provide a simple means of intuitively understanding how variability in input variables results in a distribution of forecast values.

Crystal Ball (Decisioneering Inc.)

Release date: 1993 Approximate cost: $295 Number sold: 6,000 Source language: Pascal Platforms supported: Macintosh (Lotus 1-2-3, Excel), PC/Windows (Lotus 1-2-3, Excel)

This tool has a well-designed graphical interface and a detailed, lucid manual. Spreadsheet cells designated by the user as representing uncertain input variables are called Assumptions. A Distribution Gallery allows the user to choose from among 15 standard statistical distribution types or to graphically create a custom distribution. The manual provides very clear advice about the characteristics and appropriateness of each type. Correlations among input variables can be specified by a matrix of Spearman rank correlation coefficients or can be drawn graphically. Forecast Charts, the histograms of output variable values, can be viewed dynamically during the simulation run as well as at the completion of the run. (Crystal Ball uses its own proprietary graphics engine to display these graphs, rather than that of the spreadsheet). Other analysis tools include Trend Charts for displaying and connecting confidence intervals around the forecast variable in a time-series, and Sensitivity Charts which show the importance of each uncertain input variable in determining the forecast, in terms of their rank order correlation.

@Risk (Palisade Corp.)

Release date: Approximate cost: $395 Number sold: Source language: Platforms supported: Macintosh (Excel), PC/DOS (Lotus 1-2-3), PC/Windows (Excel)

This tool makes use of the graphical capabilities of the spreadsheet in which it is embedded, with the advantage that its graphs can be easily exported to other applications. The program comes with a lengthy, well-designed manual, including an extensive tutorial with example models. Uncertain input variables can be assigned any of 38 built-in statistical distribution types. Correlations among input variables can be specified by user-provided correlation matrices. The user can generate detailed statistical reports on any output cell, including calculation of the probability of achieving a given target value. @Risk is more difficult to use than Crystal Ball but provides the user with more modeling power and flexibility.

(This page intentionally blank)

4. Research and Development Areas

We have identified a number of gaps and shortfalls in the state of the art of computer-aided business engineering and simulation. An initial discussion of these is provided in Salter and Burstein (1993). In this section of the current report, we consider the gaps and shortfalls we have identified from a different perspective: in terms of developing a preliminary research and development (R&D) agenda. We articulate the agenda in terms of the problems with existing modeling technology that could be addressed. We also provide preliminary assessments of the importance of addressing the problems and the approximate amount of difficulty that will be encountered in doing so.

We distinguish among three R&D areas, listed from the most high-level and general aspect to the most specific:

- simulation support for strategic thinking,
- the structure and content of simulations, and
- specific enhancements of simulations in the context of workflow.

In our earlier report, we identified several gaps in existing simulations that can be seen as relating to broad, **strategic issues** for an organization.

There is much talk about business process redesign (or re-engineering) in the product literature about and at seminars on organizational modeling and workflow. However, there is virtually no support in the existing technology for what Hammer and Champy (1993) define as essential to BPR: the "clean sheet of paper" approach to redesign. Existing systems tend to be inertial, facilitating making modifications to an existing design rather than developing a new design driven by higher-level goals and constraints. This "radical redesign" model can also be viewed in the context of strategic thinking, as described in a recent paper by Mintzberg (1994). Comparing strategic thinking to strategic planning (which he thinks actually inhibits true strategic change), he makes an argument quite similar to Hammer and Champy's: "[R]eal strategic change requires not merely rearranging the established categories, but inventing new ones" (Mintzberg, 1994, p. 109).

Most generally, the *de novo* **design** of a business process is strategic. The context or constraints for a design can vary widely. Highly constrained design problems are at one extreme. Examples include: designing how a unit will operate after a 10 percent cut in personnel; designing how a unit will operate with a reduction in spare parts or component inventories from one level to another; and designing how a unit should operate to support a 20 percent increase in capacity. Design problems can also be substantially less constrained, such as: designing the integrated customer service process for a business that has made a recent acquisition; designing a new sales and marketing operation to support the introduction of a new product; or redesigning the process from executing a purchase order to delivering the material to the internal customer to reduce costs and elapsed time by 50 percent each. (The last example

would fit Hammer and Champy's (1993) "clean piece of paper" definition of BPR, while the others do not.)

For designing a process, it must be possible to manipulate readily the existing design and to obtain various measures of it. The design task requires that the simulation or modeling system support a dynamic style of use where changes can be made, the model can be rerun and outputs evaluated, and the user can iterate fairly rapidly. Especially for the less-constrained design tasks, it is also important to be able to explore a large design space. That is, a user should be able to construct and evaluate very different designs, not just make incremental changes in the existing one. As we will see, these *desiderata* are substantially easier to describe than they are to find in the available systems.

Strategic thinking cuts across organizational boundaries. We identified the inability of simulations to model and capture effectively attributes of processes as they **affect different domains** in a business as a problem with existing software. Major changes -- and many of the BPR examples discussed in the literature -- cut across areas including technology; personnel issues such as required skill mixes and training requirements, performance review, and compensation policies; and management and control issues such as structures of teams, reporting, and monitoring procedures. Organizational changes can have consequences across some or all of these areas.

A related issue is that simulations have limited capabilities to deal with the **connections between the process(es) being modeled and other processes** and activities in the business. Some such limitation is inherent in the fact that any simulation must be bounded and cannot model what is beyond its boundaries. However, current simulations have virtually no capabilities to help users to consider "boundary effects," the ways in which the object of the simulation interacts with other aspects of the business. For example, a manufacturing operation must get its orders and material inputs from somewhere and send its production somewhere. (Presumably, orders come from sales, materials from suppliers and inventory; production goes to shipping.) In simulations, unless orders and inventories are explicitly being modeled, they will merely be (potentially stochastic) inputs to manufacturing. The theme runs through Hammer and Champy (1993) that one of the major virtues of BPR is that it can help managers to see connections across different parts of a business.

We believe that substantially enhancing capabilities to support strategic thinking is an open R&D area, with considerable research required before extensive development can begin. It will probably prove difficult to develop capabilities that can substantially enhance strategy development. However, there is a smaller step that could add value in dealing with some strategic issues and that is not on the research frontier. A variety of methods could be employed to assist users in identifying boundaries and potential cross-functional interactions in organizational models without modeling those in any detail. Simply providing capabilities for more extensive modeling of sources and sinks (using context-sensitive rules, for example) could be useful. These would provide some ways of labeling and presenting the results in the context of the simulation results. Thus, for example, running a

simulation to help identify bottlenecks in manufacturing would also provide explicit results on the changes in the sink for manufacturing, which would be identified as shipping, and on the changes in the required sources for manufacturing, which might be orders and supplier deliveries.

Several aspects of the **structure and content** of simulations could benefit from R&D. First, existing simulations are designed for flows. These models present the risk that by focusing on the flow, the modeler will lose sight of the work itself. Of course, flows are important, and speeding them up is one means of enhancing productivity. But the content of what gets done is also quite important, especially with the emerging interest being given to restructuring jobs: the nature and mix of tasks that individuals do.

We have identified two approaches that could help address the problem. First, a rich object-oriented approach to simulation makes it possible for different types of submodels to be used to model different simulation objects (as long as those submodels consume and produce the right sort of inputs and outputs for the submodels they communicate with). For example, some could be systems of equations, others could be simple delay and transformation mechanisms, others could be rich causal models that begin to get at the content of different kinds of jobs. This would make it possible to embed models of the content of the work within larger flow simulations. Some object-oriented simulation environments already exist, and others will be developed, as object-oriented technology continues to spread. As they do, they will begin to address these issues more directly.

The second approach is more innovative and would benefit from an active effort to support R&D. Essentially, the idea is to develop an **agent-based simulation framework with decomposable agents.** The basic objects doing work in the simulation (the "stations" in a traditional factory simulation) would be agents which can execute procedures associated with specific goals. Each agent would have a set of procedures it could execute, some of which might lead to executing other procedures. Agents would also have decision rules that they used to help decide what procedures are appropriate in a given context, given their high-level goals. Thus, each agent could be seen as a combination of a skill set and a set of preferences and intentions, where the skills are defined by the procedures that the agents can execute, the intentions by the goals that the agent has adopted, and the preferences by the decision criteria for choosing what to execute. Now, the ability to "decompose" agents into skills and to recombine those skills into new agents would provide a mechanism to investigate job restructuring, training, and related issues. Executing each procedure would consume time and perhaps other resources (including the agent's attention).

We believe that this approach is feasible based on technology now under development in advanced R&D environments, although not yet in the commercial world. It could bring a great deal of value to users of simulations, since it would greatly extend simulations' flexibility. It might also provide a suitable framework within which the structural gaps in existing technology discussed below could be addressed. (Note also that the same basic approach could be used to define agents that were machines rather than people.)

Also important in such an environment would be some amount of modeling of agent attention and fallibility. People do not necessarily have technologically supported queues that they use to routinely supply them with inputs or things to do. Thus, they can lose track of what they have been tasked to do. They can get overloaded and lose or misschedule their work. They can make mistakes that cause them to need to redo or rework and further delay their outputs. They get interrupted for all kinds of reasons, get sick, misunderstand directions, and so forth. Developing models that support these kinds of failures under different stress levels may be important to a better understanding of some of the issues that surface at least in passing in Action Technologies model of workflow cycles, where there is always the possibility of failure and recovery or loss, and there is always some "customer" for the product of the work, although there is no current simulation that models how such failures come about.

Another important problem with existing simulations and their applicability to a range of important organizational problems is their inability to **deal with global changes efficiently**. Examples of such global changes include the introduction of new technology (such as electronic mail, which changes the parameters of many internal communications processes and thus might affect many steps that depend on communications) and changes in policies (which might affect priorities among tasks, how some work gets done, etc.). The problem is that introducing a new technology or implementing a new policy might change how many separate steps are done. Currently, all those places must be found and the relevant parameters changed.

An object-oriented approach can help address this but would require that all relevant references inside steps themselves be objects. For example, all steps that would be affected by the introduction of email must have a "communicate" method within them, specialized to indicate if it can use asynchronous communication (as provided by email) or requires synchronous (as provided by telephone calls and meetings). This requires a considerable amount of discipline in constructing objects and their methods. The agent-based approach discussed above brings some of these issues to the surface more directly. Some structure could be imposed on the procedures that would be internal to the agents, perhaps by building a library of some communications and technology methods.

A related problem with existing simulations is the difficulty in having the **simulation make itself change structurally as it is running.** For example, a bank might be set up to reassign personnel from certain back-office functions to tellers when lines get too long. Similarly, it is difficult to implement models of flexible methods, to incorporate models of empowered teams of employees able to manage their own work, and the like. Such ideas are emerging as quite important in current discussions of reorganizing work, and the inability to deal with them naturally in simulations represents a substantial gap in the state of the art. A related example is modeling the effects of communications that change the structure of what is being done (by changing priorities, by reassigning personnel, by renegotiating commitments).

Taking an example from the history of simulations, Pritsker Corporation has done many studies where they act as consultant and simulation builder in which the object

of study is a set of *policies* for scheduling, such as their review of the utilization of operating rooms in a local hospital. The goal of such studies (and many similar ones in manufacturing, especially in job shop situations) is to determine a set of rules that will, in general, lead to reasonably efficient schedules for the use of the available resources. Current generation simulation environments do not support automated scheduling, as would be valuable in considering tradeoffs that might be made in such studies. In the case of the studies done with SLAMSYSTEM, the scheduling rules were implemented by the consultants as straight FORTRAN code and so were not easily changeable to review a variety of possible methods.

Similar kinds of problems occur in agile manufacturing and job-shop manufacturing. In agile manufacturing, in particular, there is a need to understand how teams of workers that can form dynamically might approach the planning and scheduling of their tasks. A modeling tool that could handle specifications of planning rules and other kinds of *policies for composing work processes* would be an important step toward being able to understand the dynamics of these kinds of organizations.

Embedding planning and scheduling behavior in a simulation requires that the simulation have an ability to "know" about itself as it is running, and to have procedures within the simulation affect its action flow structure, operating parameters, and the like. This is a suitable R&D target, which could build on existing AI-based technologies without requiring a substantial investment in cutting-edge R&D.

Finally, we have identified a specific area dealing with simulation in the context of workflow where we believe R&D could be of substantial utility. As discussed in our treatment of workflow, existing systems have only limited simulation ability. However, there is vigorous industry activity in workflow, and the simulation abilities of existing packages will undoubtedly be enhanced. (Indeed, many vendors report that they have such efforts underway.) However, we do not know of any current activity to "close the loop" in workflow, to make it possible to integrate smoothly actual data on work processes that have been collected as part of workflow management and analysis with a simulator. Most workflow systems collect data on the various work steps and make these accessible for monitoring and analysis. They also include well-specified models of the process being automated.

It would be effective to be able to use those data to parameterize steps in the workflow and then simulate alternative arrangements of the workflow, including changes in staff mixes. Simulations thus parameterized would have very high fidelity. In addition, many possible alternative configurations could be implemented fully in software. For example, in an insurance claims processing operation, analysis might reveal that the mix of credit checking and claims processing being done by a group of people should be changed. If people perform both tasks, that mix could be shifted merely by making changes in the underlying workflow. (In general, this could be done when the changes affect the mix of tasks or task priorities within people; when tasks must change across people, such changes cannot always be made purely via software.) This capability to feed operational data smoothly into simulations, which

could then be used to drive workflow modifications, would insert the mechanism for a continuous improvement cycle into workflow, substantially enhancing its utility.

5. References

Action Technologies (1993). *ActionWorkflow ANALYST User's Guide.* Alameda, CA: Action Technologies, Inc.

Galbraith, J. R. (1977). *Organization Design,* Addison-Wesley, Reading MA.

Hammer, M. , & Champy, J. (1993). *Reengineering the Corporation*. HarperBusiness Press, New York, NY.

Levitt, R. E., Cohen, G. P., Kunz, J. C., Nass, C. I., Christiansen, T., & Jin, Y. (1993). *The "Virtual Design Team": Simulating how organization structure and information processing tools affect team performance.* Stanford University Center for Integrated Facility Engineering, Tech. Report No. 93.

Lloyd, Capt. B. A., & Clark, P. K. (1993). *Object-Oriented Simulation with IMDE.* Technical Report from TASC.

Marshak, R.T. (1993) Young and Rubicam Improves Productivity with Workflow. *Workgroup Computing Report*, Vol. 16, No. 5, pp. 12-20.

Mintzberg (1994). The Fall and Rise of Strategic Planning. *Harvard Business Review*, Jan-Feb 1994, pp.107-115.

Petri, C. A. (1966). *Kommunikation mit Automaten.* Schriften des IIM Nr. 2, Institut fur Instumentelle Mathematik, Bonn. English translation is: Technical Report RADC-TR-65-337, Griffiths Air Force Base, New York, Vol. 1, Suppl. 1, 1966.

Pritsker (1989). SLAM II, Product Information. Indianapolis, Indiana.

Salter, W., & Burstein, M. (1993). CABES: Related Work and Shortfalls in the State of the Art. BBN Report No. 7923, October, 1993.

SRI International. (1993) Business Process Redesign: Improving the way your company works. Set of slides. SRI International, Menlo Park, CA

Winograd, T., & Flores, F. (1986). *Understanding Computers and Cognition.* Reading, MA: Addison-Wesley Publishing.

(This page intentionally blank)

6. Glossary

ADW	Application Development Workbench
AGV	Automatically Guided Vehicles
AI	Artificial Intelligence
AIM	Analyzer for Improving Manufacturing
AL/HRGA	Armstrong Laboratory/Human Resources Group Acquisition
AML	Activity Modeling Language
API	Application Program Interface
ART	Automated Reasoning Tool
AST	Application-Specific Templates
BPR	Business Process Reengineering
CABE	Computer-Aided Business Engineering
CABES	Computer-Aided Business Engineering and Simulation
CABRE	Computer Aided Business Re-Engineering
CAD/CAM	Computer-Aided Design/Computer-Aided Manufacturing
CASE	Computer-Aided Software Engineering *or* Computer-Aided Systems Engineering
CPN	Colored Petri-Nets
DDE	Dynamic Data Exchange
DMACS	Distribution, Manufacturing, Accounting, Costing, and Simulation
DoD	The Department of Defense
ER	Entity-Relationship
FIPS	Federal Information Processing System
GUI	Graphical User Interface
ICN	Information Control Network
ICAM	Integrated Computer-Aided Manufacturing
ICOM	Inputs, Controls, Outputs, and Mechanisms
IDEF	ICAM Definition Language
IDL	Integrated Definition Language

IMDE	Integrated Model Development Environment
ISPA	Intelligent Statistical Process Analysis
KBE	Knowledge-Building Environment
KEE	Knowledge Editing Environment
LAN	Local-Area Network
LOVEM	Line of Visibility Methodology
MIS	Management Information Systems
NC	Numeric Control
OLE	Object Linking and Embedding
OMW	Object Management Workbench
PC	Personal Computer
PERT	Program Evaluation and Review Technique
R&D	Research and Development
RDT&E	Research, Development, Training, and Evaluation
RT	Real-time
RT-DAS	Real-time Data Acquisition System
SML	Structured Modeling Language
SOW	Statement of Work
SQL	Structured Query Language
TQM	Total Quality Management
UNIX	not an acronym
VDT	Virtual Design Team
WAN	Wide-Area Network
WIP	Work-in-progress

7. Bibliography

Burstein, M., Ferguson, W., & Abrett, G. (1993). A Simulation tool for evaluating coordination strategies in organizations. Paper presented at AAAI workshop on AI Theories of Groups and Organizations: Conceptual & Empirical Research. AAAI, Washington, D.C.

Carley, K. & Lin, Z. (1993) Maydays and Murphies: A Theoretical Study of Organizational Performance under Stress. Carnegie Mellon University, Pittsburg, PA. Submitted to *Management Science.*

Cohen, M., March, J., & Olsen, J. (1972). A Garbage Can Model of Organizational Choice. *Administrative Science Quarterly* 17, pp. 1-25.

Duimering, P. R., Safayeni, F., & Purdy, L. (1993). Integrated Manufacturing: Redesign the Organization before Implementing Flexible Technology. *Sloan Management Review*, Summer, 1993, pp. 47-56.

Jensen, K. (1992). *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use.* Springer Verlag, Berlin, Heidelberg.

Malone, T. W., Crowston, K., Lee, J., & Pentland, B. (1993) Tools for inventing organizations: Toward a Handbook of organizational processes. Center for Coordination Science Working Paper No. 141, Massachusetts Institute of Technology, Cambridge, MA.

Taqi, A. A. Q., Al-Sammak, A. J., Khan, A. A., & Ahmed, N. (1992). A comparitive study between Petri Net and SLAM. In *Simulation*, November 1992. V. 59, No. 5.

(This page intentionally blank)

U. S. Government Printing Office 1995 650-075/00061